

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО**

Факультет інформатики та обчислювальної техніки
(назва факультету, інституту)

Кафедра автоматизованих систем обробки інформації і управління
(назва кафедри)

"На правах рукопису"
УДК 519.8

«До захисту допущено»
Завідувач кафедри

(підпис) О.А.Павлов
(ініціали, прізвище)

“ ” 20 19 р.

**МАГІСТЕРСЬКА ДИСЕРТАЦІЯ
на здобуття ступеня магістра**

за спеціальністю 126 Інформаційні системи та технології
(код та назва спеціальності)

ОПП Інформаційні управляючі системи та технології
(код та назва спеціалізації)

на тему: Інформаційна рекомендаційна система в сфері освітніх послуг

Виконала: студентка VI курсу групи ІС-82мп
(шифр групи)

Зінченко Людмила Вікторівна
(прізвище, ім'я, по батькові)

(підпис)

Науковий керівник с.н.с., проф., д.т.н. Гуляницький Л. Ф.
(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Консультант доц., доц., к.т.н. Жданова О.Г.
(науковий ступінь, вчене звання, прізвище, ініціали)

(підпис)

Рецензент провідний н.с. ІК НАНУ, д.ф.-м.н.,
с.н.с. Семенова Н.В.
(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Засвідчую, що у цій магістерській дисертації
немає запозичень з праць інших авторів без
відповідних посилань.

Студент

(підпис)

Київ – 2019

РЕФЕРАТ

Магістерська дисертація: 85 с., 7 рис., 23 табл., 29 джерел, 1 додаток.

Актуальність. Сьогодні є актуальною онлайн-освіта. На жаль, в Україні мало альтернативних ресурсів, де б можна було отримати онлайн-допомогу з різних предметних областей. Все більше учнів, студентів, людей, які перекваліфіковуючись чи просто хочуть розвиватися шукають способи отримати нові теоретичні знання та практичні навички онлайн. Надзвичайно важко самотійно опанувати великий потік інформації, яку б предметну область не вивчали учні, і тому необхідна допомога професіоналів. Тож великою цінністю представляється функціонал щодо розкладу. Для вирішення проблеми, як реалізувати цю частину функціоналу, буде поставлена і розв'язана нова математична задача, яка дасть можливість і підґрунтя вирішувати подібні проблеми.

Зв'язок роботи з науковими програмами, планами, темами. Робота виконувалась на філії кафедри автоматизованих систем обробки інформації та управління Національного технічного університету України «Київський політехнічний інститут ім. Ігоря Сікорського» у рамках науково-дослідницької теми Інституту кібернетики ім. В. М. Глушкова НАН України ВФ.180.11 «Розробити математичний апарат, орієнтований на створення інтелектуальних інформаційних технологій розв'язування проблем комбінаторної оптимізації та інформаційної безпеки» (2017-2021 рр.), що виконується за Постановою бюро Відділення інформатики НАН України від 23.06.2016 р. № 2.

Мета роботи – підвищення якості інформування потенційних споживачів та інтелектуалізація процесів надання освітніх послуг онлайн, шляхом розробки оригінального програмно-алгоритмічного забезпечення та реалізації його у вигляді спеціалізованої програмної системи.

Для досягнення мети необхідно виконати наступні **завдання**:

- виконати огляд існуючих постановок задач у сфері освіти;

- виконати огляд існуючих методів розв’язання задач складання розкладу;
- здійснити порівняльний аналіз різних методів та моделей та класифікувати їх;
- формалізувати задачу складання розкладу для менторів та учнів;
- розробити алгоритм локального пошуку і мурашиний алгоритм;
- виконати аналіз експериментальних досліджень;
- розробити програмне забезпечення для надання послуг у сфері освіти;
- розробити стартап-проект.

Об’єкт дослідження – процес побудови розкладу для менторів і учнів, який задовольняє певним критеріям.

Предмет дослідження – методи та моделі задач комбінаторної оптимізації в задачах теорії розкладів.

Наукова новизна одержаних результатів полягає у постановці та аналізі нової задачі, а також у дослідження методів розв’язання цієї задачі, розробці методів локального пошуку та мурашиного алгоритму для поставленої задачі складання розкладу онлайн занять.

Публікації. Матеріали роботи опубліковані в міжнародних наукових журналах «INNOVATIVE SOLUTIONS IN MODERN SCIENCE» (№6 (33), 2019), та «POLISH JOURNAL OF SCIENCE» (№16, 2019), а також у тезах міжнародних науково-практичних конференцій «Математичне та імітаційне моделювання систем» (МОДС 2019), «Інформаційні системи та технології управління» (ІСТУ-2019).

ОНЛАЙН ОСВІТА, ТЕОРІЯ РОЗКЛАДІВ, NP-СКЛАДНІСТЬ,
КОМБІНАТОРНА ОПТИМІЗАЦІЯ, МУРАШИНІ АЛГОРИТМИ, ЛОКАЛЬНИЙ
ПОШУК, МЕНТОР, УЧЕНЬ

ABSTRACT

Master's thesis: 85 p., 7 figures, 23 tables, 29 sources, 1 applications. **Relevance:** Online education is relevant today. Unfortunately, there are few alternative resources in Ukraine where online help can be obtained from various subject areas. More and more students from schools and universities, people who are retraining or just looking to develop are looking for ways to gain new theoretical knowledge and practical skills online. It is extremely difficult to master a large flow of information on your own, whatever the subject area is not learned by students, and therefore requires the help of professionals. Therefore, scheduling functionality is of great value. To solve the problem of how to implement this part of the functionality, a new mathematical problem will be posed and solved, which will give the opportunity and the basis for solving such problems.

Connection of the thesis with scientific programs, plans, topics. The thesis was written at the branch of The Department of Computer-aided management and data processing systems of the National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute» at the V. M. Glushkov Institute of Cybernetics of the National Academy of Sciences of Ukraine under the topic VF.180.11 «To develop a mathematical apparatus focused on the creation of intelligent information technologies for solving combinatorial optimization and information security problems» (2017-2021 biennium), which is executed by the Resolution of the Bureau of Informatics of the National Academy of Sciences of Ukraine from 23.06.2016 p. № 2.

The purpose of the study is improving the quality of informing potential consumers and intellectualizing the processes of providing educational services online, by developing original software and algorithmic software and implementing it in the form of a specialized software system..

To achieve this goal, you must complete the following tasks: – review the existing formulations of educational tasks; – review existing methods for scheduling tasks;

- carry out comparative analysis of different methods and models and classify them;
- formalize the timetable for mentors and students;
- develop a local search algorithm and an ant algorithm;
- carry out the analysis of experimental studies;
- develop software to provide educational services;
- develop a startup project.

The object of study is a process for scheduling mentors and students that meets certain criteria.

The subject of study is methods and models of combinatorial optimization problems in scheduling theory problems.

The scientific novelty of the results is the formulation and analysis of a new task, as well as the study of methods for solving this problem, the development of methods of local search and ant algorithm for the task of scheduling online classes.

Publications. Work materials have been published in the international scientific journals «INNOVATIVE SOLUTIONS IN MODERN SCIENCE» (№6 (33), 2019) and «POLISH JOURNAL OF SCIENCE» (№16, 2019), as well as in theses of international scientific and practical conferences «Mathematical and systems simulation» (MODS 2019), «Information systems and control technologies» (ISTU-2019).

ONLINE EDUCATION, SCHEDULE THEORY, NP-COMPLICITY, COMBINATORY OPTIMIZATION, ACO, LOCAL SEARCH, MENTOR, STUDENT

ЗМІСТ

ВСТУП.....	9
1 ОГЛЯД РІШЕНЬ У СФЕРІ ОНЛАЙН ОСВІТИ	11
1.1 Огляд наявних постановок задач	11
1.2 Огляд наявних методів розв’язання	13
2 ПОСТАНОВКА ЗАДАЧІ СКЛАДАННЯ РОЗКЛАДУ ОНЛАЙН ЗАНЯТЬ ТА РОЗРОБКА АЛГОРИТМІВ ЇЇ РОЗВ’ЯЗАННЯ	20
2.1 Змістовна постановка задачі	20
2.2 Математична постановка задачі	23
2.3 Розробка алгоритмів.....	27
2.3.1 Локальний пошук	27
2.3.2 Мурашиний алгоритм	29
3 РЕЗУЛЬТАТИ ЕКСПЕРИМЕНТАЛЬНИХ ДОСЛІДЖЕНЬ	33
4 ОПИС ПРОГРАМНОГО ТА ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ	36
4.1 Архітектура програмного забезпечення	36
4.2 Засоби розробки.....	40
4.2.1 Серверна частина.....	40
4.2.2 Клієнтська частина	47
4.3 Вимоги до технічного забезпечення.....	49
5 РОЗРОБКА СТАРТАП-ПРОЕКТУ	51
5.1 Опис ідеї проекту	51
5.2 Технологічний аудит ідеї проекту	56
5.3 Аналіз ринкових можливостей запуску стартап-проекту	58
5.4 Розроблення ринкової стратегії проекту.....	66

5.5 Розроблення маркетингової програми стартап-проекту	69
ВИСНОВКИ.....	74
ПЕРЕЛІК ПОСИЛАНЬ	76
ДОДАТОК А	79
Структура термінів для сфери навчання.....	80
Візуалізація постановки задачі	81
Схематичне зображення вхідних і вихідних даних задачі.....	82
Процес визначення однодумців	83
Процес визначення важливості предметної області	84
Псевдокод алгоритму знаходження одного варіанту розкладу.....	85
Структура графу даної задачі для мурашиного алгоритму	86

ВСТУП

В умовах глобалізації та діджиталізації актуальною стає онлайн освіта. Ми живемо в інформаційному суспільстві, у світі, у якому все змінюється з величезною швидкістю, у якому кожній людині щодня доводиться опрацьовувати великі потоки інформації. Багато процесів автоматизуються, і отримання освіти не є виключенням. Нині у всьому світі здійснюється автоматизація підприємств, і в Україні також. У зв'язку з цим, багато людей втрачають роботу і змушені отримувати ще одну кваліфікацію, і оскільки часу і грошей на отримання нової освіти у ВНЗ немає, краще зайнятися самоосвітою.[1]

Тисячі людей щодня не тільки перекваліфіковуються, а і паралельно з чинною роботою прагнуть освоїти ще одне або декілька занять (чи підняти рівень знань в області, що безпосередньо їх торкається кожного дня), чи то для власного розвитку, чи для більшого заробітку, чи від поклику серця або від нудьги. Найлегше це зробити в Інтернеті. Проте не завжди є відповідні ресурси для цього.[1]

Так, наприклад, учні, які роблять домашні завдання, чи студенти, які виконують лабораторні та практичні завдання в університетах, потребують допомоги, людину, яка пояснить їм незрозумілий матеріал [1]. Для цього є репетитори, наставники, коучі або, як будемо називати їх ми, ментори. Ментор – це людина, що має теоретичні чи практичні знання будь-якого рівня у певній предметній області, виявляє бажання ділитися своїми знаннями та навчати інших в тій чи іншій формі, допомагає навчатися та консультує у питаннях, які відповідають його предметній області[1]. Ментор – це людина, яка самостійно встановлює інтенсивність та напрямок навчання. Завдання ментора – допомогти тоді, коли потрібна допомога. Питання в тому, як знайти менторів? Ще декілька десятиліть тому пошук здійснювався по оголошеннях або через знайомих, проте нині пошук здійснюється у більшості випадків через Інтернет-ресурси.

Нині в Україні існує дуже мало Інтернет-ресурсів, які надають можливість викладачам та менторам публікувати безкоштовні оголошення про готовність надавати допомогу. Ці нечисленні наявні Інтернет-ресурси мають ряд недоліків, як от: незручний та несучасний інтерфейс, мало функціоналу, заточені під одну справу.

Метою роботи є підвищення якості інформування потенційних споживачів та інтелектуалізація процесів надання освітніх послуг шляхом розробки оригінального програмно-алгоритмічного забезпечення та реалізації його у вигляді спеціалізованої програмної системи.[1]

Досягнення мети базується на розробці оригінальних математичних методів та алгоритмів та їх реалізації у спеціальній системі, яка призначена для формування та модифікації чи уточнення розкладів занять учні з менторами; окремі блоки подібної системи були розроблені автором (разом з Костичевою К. Ю.) в рамках бакалаврської роботи.

Отже, на сьогодні онлайн-освіта є напрочуд актуальною. Все більше учнів, студентів, людей, які перекваліфіковуючись чи просто хочуть розвиватися шукають способи отримати нові теоретичні знання та практичні навички онлайн. Надзвичайно важко самотійно опанувати великий потік інформації, яку б предметну область не вивчали учні, і тому необхідна допомога професіоналів. Система полегшить пошук професіоналів, тобто менторів, які мають бажання і допомагатимуть у певній предметній області. Зі свого боку ментори мають простір для творчості та креативності, можливість проявити себе як викладач та наставник і допомогти учням та студентам отримати нові знання та заповнити прогалини у вже наявних. Тож великою цінністю представляється функціонал щодо розкладу. Для вирішення проблеми, як реалізувати цю частину функціоналу, буде поставлена і розв'язана нова математична задача, яка дасть можливість і підґрунтя вирішувати подібні проблеми.

1 ОГЛЯД РІШЕНЬ У СФЕРІ ОНЛАЙН ОСВІТИ

У сфері освіти наявні задачі, що зазвичай належать до теорії розкладів. Проте якщо мова йде про онлайн-освіту, то постановка задачі може відрізнятися від класичних. Отже, дослідимо наявні постановки задач, класифікуємо їх та методи їх розв'язування. На основі цього аналізу зможемо визначити актуальність формулювань нових постановок задач або видозмінення існуючих для врахування різних вимог проектів, які стосуються онлайн-освіти. Відзначимо, що близькими до задач, на яких акцентується розгляд, та алгоритмів їх розв'язування є задачі та підходи до їх розв'язування, викладені рядом авторів, зокрема, в [2-5].[1]

1.1 Огляд наявних постановок задач

У загальному формулюванні задача складання розкладу полягає в наступному: використовуючи деяку множину ресурсів або обслуговуючих пристроїв, повинна бути виконана деяка фіксована множина завдань із заданим набором характеристик. У багатьох випадках розглядаються такі характеристики як: перелік операцій, які входять до складу кожного із завдань; обмеження на послідовність виконання операцій кожного завдання й часткові послідовності термінів їх завершення, а також терміни виконання завдань; ресурси, які необхідні для виконання кожної з операцій усіх завдань; час та вартість виконання робіт кожної із операцій завдань при використанні різних видів ресурсів; директивні терміни початку та завершення виконання деяких чи всіх завдань; перелік та характеристики ресурсів, необхідних для виконання всіх операцій кожного із завдань.[2]

Коло розв'язуваних у рамках теорії розкладів задач можна розділити на наступні групи:

- задачі розподілу завдань, вибір ресурсів (пристроїв) для їх виконання;
- задачі упорядкування виконання множини завдань на кожному із пристроїв;

- задачі узгодження часу виконання виділених підмножин завдань, що виконуються;
- задачі вибору складу та розміщення обладнання, визначення кількості, об'ємів та розміщення проміжних буферних накопичувачів;
- маршрутизація руху виробничого транспорту;
- побудова розкладів проведення занять, руху автомобільного, залізничного, міського транспорту та руху літаків.[3]

Задача, яка розглядається у даній роботі, належить до останньої групи. Багато науковців приділяють увагу задачам побудови розкладів проведення занять. Наприклад, у [4] запропоновано розроблення математичної моделі розкладу занять у ВНЗ, яка дозволяла б ефективно розв'язувати задачу автоматичного складання розкладу занять і була б гнучкою у випадку адаптації системи для розв'язання конкретної практичної задачі. Пріоритетне право викладачів на розподіл занять в алгоритмі складання розкладу визначається за методом ранжування посад. У [5] розглядається метод автоматизованого створення розкладу, що забезпечує вимоги Болонського процесу щодо складання індивідуальних планів магістрантів. Класична задача навчального планування оперує поняттями учень, група, викладач, заняття, аудиторія.

Відмінність наукових робіт на цю тематику полягає у різних обмеженнях відносно вищевказаних понять. Можна виділити такі:

- один викладач у кожен момент часу проводить не більше одного заняття [6-9];
- у кожен момент часу в одній аудиторії може проводитися не більше одного заняття [6-9];
- у однієї навчальної групи у кожен момент часу може бути не більше одного заняття [6-9];
- умова відсутності “вікон” для навчальних груп [10,11];
- обмеження на максимальну кількість пар для учнів, викладача [10];

- відповідність типу аудиторії і виду заняття [10];
- часові обмеження на день [12];
- пріоритетне проведення певного типу занять (наприклад, лекцій) на перших парах [12];
- врахування побажань викладачів [8,12];
- зменшення переходів між аудиторіями [8,12].

У релевантних дослідженнях подібні обмеження можуть комбінуватися, крім того, також може вводитися пріоритизація обмежень [13].

1.2 Огляд наявних методів розв'язання

Задача складання розкладу навчальних занять є NP-складною, тобто такою, для якої невідомий алгоритм її розв'язування із поліноміальною складністю в залежності від розмірності. Задачі теорії розкладів мають широке застосування, наприклад, при оптимізації роботи підприємств, де виконується велика кількість операцій в різні проміжки часу, чи при оптимізації процесів обробки великих обсягів інформації.

Серед приближених методів теорії розкладів виділяють групу евристичних алгоритмів. В цих методах так чи інакше відтворюється досвід, накопичений в результаті багаторазової побудови розкладів в практично однотипних умовах. Область пошуку найкращого розкладу при цьому штучно звужується, часто до одного варіанту розкладу. [15]

Багато задач теорії розкладів зводяться до задач комбінаторної оптимізації (ЗКО).

Індивідуальною задачею оптимізації називається пара $(f(x|c), X)$, де $f(x|c)$ – цільова функція, c – набір даних задачі (вхід задачі), а X – простір розв'язків задачі (простір пошуку) [14].

Під задачею оптимізації розуміють сукупність усіх можливих індивідуальних задач [14].

Залежно від типу простору розв'язків X розрізняють задачі неперервної, комбінаторної та змішаної оптимізації [16]. Перший клас утворюють задачі, у яких простір X є неперервним (континуальним). Щодо другого класу, то тут існують різні підходи до означення як самого поняття ЗКО, так і його підкласів – власне комбінаторних задач, а також задач дискретного й цілочислового програмування [14].

Існує багато підходів до класифікації алгоритмів комбінаторної оптимізації (АКО) – за точністю, типом використаних просторів, структурою обчислювальної схеми тощо [14].

За складністю структури АКО можна виділити [14]:

- прості алгоритми;
- комбіновані алгоритми;
- метаевристики;
- гібридні метаевристики;
- гіперевристики.

Комбіновані алгоритми утворюються шляхом послідовного застосування двох чи більше ітераційних алгоритмів з передаванням розв'язків від одного до іншого. У метаевристиках, про що піде далі, здійснюється вкладення одного алгоритму/процедури в іншу стратегію [14].

Гіперевристикою (гіперевристичним алгоритмом) називають метод пошуку, орієнтований на автоматизацію процесів вибору, комбінування або адаптації чи налаштування кількох простіших алгоритмів (евристик або метаевристик) для ефективного розв'язання ЗКО чи їх класів. Це може досягатися як вибором наявних евристик чи їх фрагментів, так і генеруванням нових [14].

Таким чином, якщо метаевристики та інші алгоритми здійснюють переважно пошук у просторі розв'язків ЗКО, то простором пошуку для гіперевристик є множина евристик (простіших алгоритмів чи їх частин) [14].

За впливом на ландшафт пошуку більшість АКО можна віднести до таких, що залишають його незмінним. Проте є алгоритми, які модифікують цей ландшафт шляхом [14]:

- зміни простору розв'язків (наприклад, послідовні алгоритми);
- зміни цільової чи оцінкової функції (алгоритми керованого локального пошуку);
- варіації системи околів, що використовується при пошуку (алгоритми локального пошуку (ЛП) зі змінними околами, метод вектора спаду з пульсуючими околами).

Якщо робота алгоритму базується на безпосередніх даних ЗКО, то такі АКО належать до *задаче-орієнтованих* алгоритмів. [14]

У деяких нових АКО використовуються не стільки прямі дані ЗКО, скільки спеціальна модель задачі, що розв'язується (наприклад, феромонна матриця та матриця маршрутів у ОМК), – такі алгоритми отримали назву *моделе-орієнтованих*. [14]

Точні алгоритми можна поділити на загальні методи та спеціальні алгоритми. [14]

Загальні методи [14]:

- повний перебір (вичерпний пошук);
- метод гілок і меж (МГіМ);
- метод гілок і відтинань;
- послідовний аналіз варіантів (ПАВ, "київський віник");
- динамічне програмування (метод Беллмана).

Спеціальні алгоритми будуються на основі врахування специфіки конкретної задачі оптимізації, що розв'язується, тому мають вузький спектр застосування. [17]

Далі розглянемо наближені АКО, бо саме вони, у більшості випадків, застосовуються при розв'язуванні практичних ЗКО (табл.1.1).

Далі піде мова про деякі групи наближених алгоритмів.

Генетичні алгоритми. Базуються на тому, що на кожному кроці обчислень розглядається множина розв'язків (популяція) і певним чином вибираються пари розв'язків, які рекомбінуються, і таким чином створюються нащадки.

Таблиця 1.1 – Класифікація основних наближених методів комбінаторної оптимізації

№	Загальна назва алгоритму	Назва алгоритму
1	Послідовні алгоритми	Конструктивні (у т.ч. “жадібні”)
		Алгоритми на базі точних методів (МГіМ, ПАВ тощо)
		GRASP
2	Детермінований ЛП	Стандартний ЛП
		Пошук зі змінюваними околами
		Керований ЛП
		Табу-пошук
3	Стохастичний ЛП	Алгоритми імітаційного відпалу
		G-алгоритми
		Повторюваний ЛП
		Квантовий відпал
4	Еволюційні алгоритми	Генетичні алгоритми
		Міметичні алгоритми
		Імунні алгоритми
		Диференціальна еволюція
5	Ройові алгоритми	Мурашині алгоритми
		Оптимізація потоком частинок
		Бджолині алгоритми
		Алгоритми світлячків
6	Методи сканування	Розсіяний пошук (Scatter Search)
		Перекомпоновка маршрутів
		МДБ

		Н-алгоритми
7	Спеціальні методи	“Іди в найближче місто”
		Алгоритм Ліна-Кернігана
		ϵ -наближені алгоритми

Цим створюється розширена популяція, із якої за правилами селекції формується наступна популяція. [16] Таким чином, можна розділити алгоритм на три етапи [16]:

Етап 1. Генерація проміжної популяції шляхом відбору поточного покоління.

Етап 2. Схрещування особин проміжної популяції шляхом кросовера, у результаті чого формується нове покоління.

Етап 3. Мутація нового покоління.

Для кожної задачі, що розв'язується, необхідно визначити функцію пристосованості та спосіб кодування, в іншому алгоритм універсальний і етапи алгоритму для всіх задач виконуються однаково. Не рекомендується використовувати ці методи, коли необхідно знайти всі розв'язки задачі, а не один із них. Мінусом цих алгоритмів є те, що вони ігнорують специфіку задачі, а час розрахунку функції пристосованості часто буває достатньо великим. [16]

Мурашині алгоритми. Алгоритм базується на поведінці мурах, що шукають шлях від мурашника до їжі. Проте для кожного конкретного типу задач потребують визначення параметрів, що можуть бути отримані лише експериментально. [16]

Метод імітаційного відпаду є одним із представників поширеного класу методів стохастичного локального пошуку. Характеризується тим, що при пошуку оптимального розв'язку на кожному кроці має ймовірність переходу у стан з більшим значенням цільової функції (для задач на мінімізацію), що дозволяє виходити з локальних оптимумів. Чим більша ітерація алгоритму, тим інтенсивніше зменшується ця ймовірність на кожному кроці. [16]

Серед наближених оптимізаційних методів розв'язання ЗКО окремий клас утворюють *метаевристики*. За своєю природою метаевристики об'єднують простіші алгоритми чи техніки в межах обчислювальних схем вищого рівня, які спрямовані на ефективне вивчення простору пошуку. Точніше, метаевристика — це метод розв'язання широкого класу обчислювальних задач шляхом такого комбінування існуючих процедур, при якому одна є провідною, а інша (чи кілька) - підлеглою (підлеглими). Як провідними, так і підлеглими процедурами часто стають відомі евристики чи алгоритми, зазвичай рандомізовані. Якщо складовою метаевристики є певний математичний метод, то вживають термін *метаевристика*. [14]

Серед еволюційних методів, окрім генетичних алгоритмів, які призначені для розв'язання ЗКО, визначених у бінарних просторах і комбінаторних просторах зі складнішими конфігураціями (наприклад у просторі перестановок), виділяють ще метаевристики на основі штучних імунних систем і методи диференціальної еволюції. [14]

Пошук концепцій для розробки алгоритмів комбінаторної оптимізації зумовив появу в останній час нових класів алгоритмів, які нав'язані природою. Помітне місце серед таких алгоритмів займають методи ройового інтелекту. Найпоширеніші серед таких обчислювальних схем - оптимізації мурашиними колоніями (Ant Colony Optimization) та потоком частинок (Particle Swarm Optimization), бджолині алгоритми й методи штучних бджолиних колоній (Bees Algorithms, Artificial Bee Colony). [14]

Висновки. Серед наявних постановок задача, що буде розглянута далі, є унікальною, оскільки не потребує наявності та аналізу груп і аудиторій, а матиме такі принципові параметри, як коефіцієнт схожості ментора та учня, а також коефіцієнт важливості предметної області для ментора. Коефіцієнт схожості в цьому випадку заміняє ранжування посад, що описано в постановці [4]. Оскільки змістовна постановка задачі є унікальною, то актуальною є побудова математичної моделі для цієї задачі.

Серед вище описаних методів я надаю перевагу евристичним алгоритмам, оскільки вони характеризуються стійкістю до потрапляння в локальні оптимуми і гарантують отримання деякого варіанту рішень за конкретний час. Евристичні алгоритми часто застосовуються при розв'язуванні практичних задач і дозволяють отримати відносно непоганий розв'язок задачі при відносно невеликому об'ємі обчислень [15]. Тому планую використати мурашиний алгоритм розв'язування задачі, а для пошуку початкових даних – метод локального пошуку.

2 ПОСТАНОВКА ЗАДАЧІ СКЛАДАННЯ РОЗКЛАДУ ОНЛАЙН ЗАНЯТЬ ТА РОЗРОБКА АЛГОРИТМІВ ЇЇ РОЗВ'ЯЗАННЯ

2.1 Змістовна постановка задачі

Визначимо основні терміни, що будуть зустрічатися в розділі.

Предметна область – предмети, які будуть викладатися/вивчатися на заняттях учнів та менторів. Ментор у своєму профілі може вказати як предметну область, так і предмет. І предметна область, і предмет будуть відображатися і використовуватися в системі. Предметна область ширше поняття, ніж предмет. Можна прослідкувати ієрархію цих двох термінів на рисунку А.1.

Ментор – це людина, що має теоретичні чи практичні знання певного рівня у визначених предметних областях, виявляє бажання ділитися своїми знаннями та навчати інших в тій чи іншій формі, допомагає іншим навчатися та консультує у питаннях, які відповідають його предметній області, а також самостійно встановлює інтенсивність та напрямок навчання. Отже, припускається, що конкретний ментор може мати досвід у декількох предметних областях із вказанням їх пріоритету. [16]

Однодумці – група людей зі схожими поглядами на життя, переконаннями, схожим стилем життя або люди, що займаються однією справою [16]. Термін “однодумці” можна сприймати в різному сенсі. Можливо навіть для кожного він свій. Ми ж вважатимемо за однодумців, у широкому сенсі цього слова, людей зі схожими поглядами на життя, схожим стилем життя, але це люди необов’язково з однаковими уподобаннями та рисами характеру чи темпераментом. Люди схожі, якщо мають однакові погляди на життя, а отже, спосіб мислення і стиль мислення людей подібні (якоюсь мірою схожі/однакові).

Рекомендаційна система – модуль програми, який надає можливість із певною періодичністю надсилати учням повідомлення про рекомендацію менторам учнів з певними годинами занять, а учням – менторів, враховуючи побажання всіх учнів по заняттях. Основна ідея рекомендаційної системи

показана на рисунку А.2. Вхідні і вихідні параметри цієї системи проілюстровані на рисунку А.3.

Розклад ментора – календар на кожен день погодинно з вказаними зайнятими чи вільними годинами цього ментора.

Загальний розклад – розклад, що буде створювати рекомендаційна система, він складається з розкладів усіх менторів та побажань учнів. При формуванні загального розкладу необхідно враховувати умови кожного ментора та учня, щоб не було так званих “накладок”.

Зафіксований розклад – години в календарі, що ментор позначив як зайняті шляхом підтвердження заняття з учнем, або вказавши, що він не працює в цей час.

Заявка про допомогу – зафіксоване побажання учня щодо заняття з ментором.

Перейдемо до змістовної постановки задачі.

Маємо менторів. Кожен ментор має розклад:

- статичний – той, який уже зафіксований;
- динамічний – той, який формує рекомендаційна система на основі побажань учнів з урахуванням можливостей і характеристик ментора.

Маємо учнів. Нехай всі уроки проводяться онлайн і не витрачається час на дорогу, визначення місця заняття тощо. Кожен учень створює заявку про допомогу, яка включає:

- предметну область, по якій проводитиметься заняття;
- годину, о котрій заняття починається;
- тривалість в 1 годину;
- допустимий зсув часу в 1 годину.

При цьому учні і ментори між собою можуть бути однодумцями. Кожному учневі може відповідати набір менторів, з яких ментор може краще чи гірше підходити на роль однодумця, що відображається у коефіцієнті схожості ментора з учнем. Будемо вважати, що коефіцієнт схожості знаходиться в інтервалі від 0

до 1. Якщо коефіцієнт схожості ментора з учнем приймає значення 1, то ментор і учень 100-відсоткові односторонні, якщо приймає значення 0 – не односторонні. Процес знаходження коефіцієнта схожості проілюстровано на рисунку А.4.

Потрібно врахувати побажання всіх учнів по заняттях і розподілити години між менторами так, щоб за можливості надати учневі бажаний часовий інтервал, а якщо це неможливо, то максимально врахувати його можливі зсуви. При цьому рекомендаційна система повинна врахувати відповідність предметної області ментора предметній області учня, і також врахувати вже наявний зафіксований розклад менторів. Процес визначення важливості предметної області для ментора проілюстровано на рисунку А.5.

Критерієм, який буде визначати добре чи погано рекомендаційна система склала розклад, буде максимум суми коефіцієнтів схожості врахованих учнів та залучених менторів по всім предметним областям та суми коефіцієнтів важливості предметної області для менторів стосовно назначених на них заявок про допомогу.

З алгоритмічної точки зору нам буде зручніше оперувати із в деякому сенсі коефіцієнтом “несхожості”, що утворюється шляхом віднімання від 1 коефіцієнта схожості кожної пари “ментор – учень”, а також коефіцієнтом “неважливості”, що утворюється шляхом віднімання від 1 коефіцієнта важливості кожної пари “ментор – предметна область”.

Розроблений математичний апарат для розв'язування подібних задач теорії розкладів буде використано при створенні спеціалізованої системи, зокрема, її найважливішої підсистеми «Інформаційна рекомендаційна система в сфері освітніх послуг». Створена система не матиме аналогів сьогодні на ринку ні в Україні, ні за кордоном за своєю новизною і проблемами, які були вирішені під час реалізації системи. Створювана система призначена для надання можливості спілкуватися учням та менторам через чат в режимі онлайн, всі дії між ментором і учнем будуть виконуватися без посередників, у вигляді адміністраторів та модераторів, що дуже важливо, адже цей факт виводить систему на новий рівень

послуг, що надаються в порівнянні з аналогами. У системі буде доступний не просто пошук менторів з предметної області, а й пошук по завданню чи проблемі, яку учень не може вирішити або зрозуміти, таким чином системою повністю забезпечити можливість знайти ментора, який є компетентним саме у конкретному питанні учня. Щоб зрозуміти, наскільки кваліфікований ментор, запроваджено систему оцінювання роботи менторів з боку учнів, а також надана можливість менторам публікувати статті з різних проблемних питань та областей, щоб після прочитання статі учні змогли зрозуміти, які конкретні сфери ментор розуміє, наскільки цікавим та потрібним речам ментор може навчити.

Також система дає можливість знайти однодумців на основі власних уподобань, залучати нових знайомих і спілкуватися з ними, таким чином заводячи друзів-менторів. Для пошуку однодумців було запроваджено спеціальний механізм, який знаходить однодумців, беручи до уваги різні переваги користувачів, використовуючи математичні методи. Ментори-однодумці будуть враховані в підсистемі «Інформаційна рекомендаційна система в сфері освітніх послуг».

Підсистема «Інформаційна рекомендаційна система в сфері освітніх послуг» міститиме розклад, який зможуть вести ментори та учні прямо в даній системі, щоб ефективно розподіляти свій час. Використовуючи систему, учень заходить і створює заявку, в якій каже, що хочу позайматися по такому предмету о такій-то годині, а у менторів при цьому є розклад власних занять (календар, де видно вільні вони чи ні). Система розв'язує задачу, яка полягає в тому, щоб знайти ментора-однодумця за певними критеріями, який зможе провести заняття в певний проміжок часу.

2.2 Математична постановка задачі

Є множина менторів, що навчають одному чи декільком предметам; учні подають заявки на вивчення предмету, який відноситься до однієї із областей знань. Таким чином, маємо множини учнів, менторів, предметних областей,

заявок на навчання та часові інтервали, в яких ментори доступні для навчання. Перенумеруємо кожну з цих множин. [16]

Введемо ряд позначень:

P_n – множина предметних областей;

N – кількість предметних областей;

P_m – множина менторів;

M – кількість менторів;

P_s – множина учнів;

S – кількість учнів;

P_r – множина заявок від учнів про допомогу;

R – кількість заявок;

P_t – множина часових інтервалів;

T – кількість часових інтервалів.

Отже, до k -ї предметної області, $k \in P_n$ відноситься певна кількість менторів m_{kj} , $j = 1, \dots, M$, j – номер ментора. І навпаки, кожен j -й ментор може мати відношення до декількох предметних областей із вказанням їх пріоритету у вигляді строгого ранжування, унікального для кожної k -ї предметної області, де максимально можливе значення – це кількість таких предметних областей N . [16]

Припускається, що раз на добу будуть опрацьовуватися нові заявки учнів про допомогу, що з'явилися за останню добу, а також ті, що були взяті раніше, але не підтверджені учнем та ментором. Учні, що не мають заявок, до розгляду, не братимуться до уваги. [16]

Маємо, що i -й учень, $i = 1, \dots, S$, може створити певну кількість запитів на навчання із певної предметної області. [16]

Позначимо v_{ij} – коефіцієнт схожості між i -м учнем та j -м ментором, значення якого знаходяться в межах $0 \leq v_{ij} \leq 1$, $i = 1, \dots, S$, $j = 1, \dots, M$. Тут 1 означає, що ментор та учень максимально схожі за вибором предметної області, 0 – навпаки. [16]

Пронумерувавши предметні області j -го ментора, отримаємо пріоритет кожної предметної області, що його стосується. Відобразимо ці пріоритети на відрізок $[0,1]$ так, що 1 означатиме, що дана предметна область являється найвищою за пріоритетом для ментора, а 0 – найнижчою. [16]

Нехай w_{kj} – коефіцієнт важливості k -ї предметної області для j -ого ментора, який буде задовольняти нерівності [16]:

$$0 \leq w_{kj} \leq 1, \quad k = 1, \dots, N, \quad j = 1, \dots, M.$$

Використовуючи введені поняття, сформуємо розклад ментора. Подамо обмеження на розклад ментора у вигляді матриці Y розмірності $T \times M$, елемент y_{tj} якої визначається так [16]:

$$y_{tj} = \begin{cases} 1, & \text{якщо в } t\text{-й часовий інтервал ментор зайнятий,} \\ 0, & \text{в іншому разі} \end{cases}$$

де t – часовий інтервал, $t = 1, \dots, T$,

j – номер ментора, $j = 1, \dots, M$.

Для подання розв'язку введемо матрицю X , розмірності $M \times T$, елемент x_{tj} якої визначається так [16]:

$$x_{tj} = \begin{cases} r, & \text{якщо в } t\text{-й часовий інтервал заявка призначена } j\text{-му ментору, } r \in P_r, \\ -1, & \text{якщо ментор зайнятий в } t\text{-й часовий інтервал,} \\ 0, & \text{в іншому разі} \end{cases}$$

де t – часовий інтервал, $t = 1, \dots, T$,

j – ментор, підібраний на заявку певного учня в даному варіанті розкладу,
 $j = 1, \dots, M$.

Для матриці X виконується наступні обмеження. [16]

Одна заявка повинна бути назначена лише один раз одному ментору [16]:

$$\exists t_1, t_2 \in \{1, \dots, T\} : \forall j_1, j_2 \in \{1, \dots, M\}, x_{t_1 j_1} \neq 0 : x_{t_1 j_1} = x_{t_2 j_2}.$$

Кожна заявка повинна бути оброблена [16]:

$$\forall r \in P_r \Rightarrow \exists (t \in \{1, \dots, T\}, j \in \{1, \dots, M\}) : x_{tj} = r.$$

Заявка може бути назначена ментору лише у тому випадку, якщо ментор вільний у період доби, вказаний у заявці про допомогу [16]:

$$\forall y_{tj} \in V, y_{tj} = 0 \Rightarrow \exists (t \in \{1, \dots, T\}, j \in \{1, \dots, M\}) : x_{tj} = r.$$

Нехай маємо матрицю заявок C розмірності $2 \times R$. Перший рядок цієї матриці буде містити номери учнів, що подали відповідну заявку. Другий рядок цієї матриці буде визначати предметну область, до якої належить ця заявка. Для спрощення припускаємо, що одна заявка може мати відповідати лише одній предметній області. У подальших дослідженнях ця умова може бути заміненна чи розвинена. [16]

Таким чином маємо наступні елементи матриці C : у стовпчику j елемент c_{1j} визначає номер учня, який створив заявку з номером j , а c_{2j} – предметну область, до якої відноситься ця заявка. [16]

Тоді цільову функцію задачі можна буде подати так [16]:

$$F(X) = \sum_{t=1}^T \sum_{j=1}^M \left(\alpha(1 - v_{c_{1,x_{tj}} j}) + (1 - \alpha)(1 - w_{c_{2,x_{tj}} j}) \right) \rightarrow \min,$$

де α – ваговий коефіцієнт, який балансує фактори схожості і важливості,

$v_{c_{1,x_{tj}} j}$ – коефіцієнт схожості між учнем $c_{1,x_{tj}}$ та j -м ментором,

$w_{c_{2,x_{ij}}j}$ – коефіцієнт важливості $c_{2,x_{ij}}$ -ї предметної області для j -го ментора.

Отже, для знаходження розв’язку поставленої задачі потрібно знайти найкращий варіант розкладу. [16]

Формалізуємо алгоритму знаходження одного варіанту розкладу у вигляді псевдокоду (рис. А.6).

2.3 Розробка алгоритмів

2.3.1 Локальний пошук

Одним з найвідоміших і найпростіших, проте потужних, евристичних алгоритмів є локальний пошук. Спробуємо застосувати цей алгоритм для пошуку найкращого варіанту розкладу. Точкою в нашому випадку буде варіант розкладу, а околom точки – модифікація цього розкладу. [16]

Специфіка поставленої задачі полягає у тому, що на точку (варіант розкладу) накладено багато обмежень, тому модифікація варіанту розкладу повинна відбуватися за певними правилами. [16]

Я вивела два правила знаходження околу точки. [16]

Допустимим ментором вважаємо такого, який має таку предметну область, що вказана в заявці, та який вільний в проміжку ± 1 година від вказаного часу в заявці. [16]

Правило 1. [16]

Крок 1. Взяти заявку випадковим чином.

Крок 2. Визначити назначеного ментора на цю заявку.

Крок 3. Визначити допустимих менторів для заявки.

Крок 4. Назначити заявці одного з допустимих менторів, відмінного від того, що зараз назначений у розкладі.

Правило 2. [16]

Крок 1. Взяти 2 заявки випадковим чином.

Крок 2. Визначити для 1-ї заявки, чи є допустимим ментор 2-ї заявки. Якщо так – крок 3. Якщо ні – вибрати випадковим чином ще одну заявку і повторити поточний крок.

Крок 3. Поміняти заявки місцями.

Оскільки незначна зміна в розкладі може призвести до різкого збільшення значення цільової функції, вважаю недоцільним брати за критерій завершення умову відсутності поліпшуючої точки в околі поточного розв'язку, що забезпечило б швидке завершення роботи алгоритму через високу ймовірність виконання цієї умови, використовуючи будь-яке з двох правил. [16]

Тому критерієм завершення оберемо досягнення певної кількості ітерацій. [16]

Застосуємо виведені правила для процедури генерації сусідніх точок. Формалізуємо алгоритм локального пошуку, використовуючи правило 1. [16]

Крок 1. Обирати початкову заявку випадковим чином.

Крок 2. Для обраної заявки знайти всіх допустимих менторів.

Крок 3. Випадковим чином обрати 1 ментора з допустимих і назначити йому заявку на час вказаний у заявці, якщо цей час зайнятий, то зі зсувом ± 1 годину.

Крок 4. Позначити заявку як оброблену.

Крок 5. Перейти до наступної необробленої заявки, вибраної випадковим чином. Повторити крок 2-4 для цієї заявки. Виконувати, поки всі заявки не будуть назначені менторам.

Крок 6. Порахувати цільову функцію (ЦФ) для складеного варіанту розкладу.

Крок 7. Обрати випадковим чином заявку, для якої допустимих менторів більше ніж 1.

Крок 8. Обрану заявку назначити допустимому для неї ментору, вибраного випадковим чином із наявних, при цьому для поточного назначеного ментора звільнити час у розкладі.

Крок 9. Порахувати ЦФ для створеного варіанту розкладу.

Крок 10. Порівняти ЦФ для розкладу, створеного на кроці 6 і на кроці 9. Якщо на кроці 9 значення ЦФ менше, то записати варіант розкладу і значення ЦФ, як поточні найкращі значення.

Крок 11. Повторювати кроки 8-10, поки не переберемо всіх допустимих менторів заявки.

Крок 12. Повторювати кроки 7-11, поки не настане умова завершення алгоритму.

2.3.2 Мурашиний алгоритм

Мурашині алгоритми – це багатоагентні системи, де поведінка кожного агента, який називатимемо штучною мурахою, або надалі просто мурахою, заснована на поведінці справжніх мурашок. Вони застосовуються для розв'язання багатьох типів оптимізаційних задач.[13] Надалі мурашині алгоритми будемо називати алгоритмами оптимізації мурашиною колонією (ОМК; Ant Colony Optimization – ACO).

Алгоритми ОМК застосовуються до задач оптимізації, які можуть бути охарактеризовані таким чином: їх розв'язок складається з компонентів, з яких можна покроково будувати фрагменти розв'язків, а на завершальному етапі – і весь розв'язок.[13]

Оскільки у ОМК модель задачі подається у вигляді графу, представимо і нашу задачу у вигляді графу. Маємо два типи вершин: заявки і ментори. Перехід від вершини-заявки до вершини-ментора можливий якщо ментор допустимий для даної заявки. Перехід від вершини-заявки до іншої вершини-заявки визначається випадковим чином і може бути лише один. Граф буде формуватися динамічно, що означає, що перехід від однієї вершини-заявки до іншої буде формуватися лише після переходу вершина-заявка – вершина-ментор, при цьому

вершина-заявка, до якої переходять, буде обиратися випадковим чином серед досі необраних вершин-заявок, як на рисунку А.7.

Специфіка нашої задачі наступна: оскільки є вимога, щоб всі заявки були оброблені, то мураха повинна пройти по всім вершинам-заявкам, обираючи на кожній вершині ментора, використовуючи при цьому феромони.

Правила оновлення феромону. Поєднаємо оновлення феромону з одночасним випаровуванням. Тоді оновлення феромону на дузі (i, j) графа задачі буде виражатися у вигляді фіксованої величини:

$$\tau_{ij}(t+1) = \lambda \tau_{ij}(t) + (1 - \lambda) \tau_0,$$

де τ_0 – початкова кількість феромону,

λ – параметр алгоритму, що називається коефіцієнтом випаровування, $0 \leq \lambda \leq 1$.

Покладемо $\tau_0 = \frac{1}{F_{\min}^0}$, де F_{\min}^0 – найкраще значення цільової функції на початковій популяції мурах, візьмемо це значення з попередньої прогонки алгоритму локального пошуку.

Правило переходу до нової вершини. Імовірнісне правило переходу мурахи від поточної вершини i до нової j вершини базуватиметься на визначенні ймовірності p_{ij} :

$$p_{ij} = \frac{b_{ij}^q \tau_{ij}^\rho}{\sum_{s=0}^S \sum_{m=0}^M b_{sm}^q \tau_{sm}^\rho},$$

де b_{ij} – бажаність i -го переходу,

τ_{ij} – кількість феромонів на i -му переході,

q, ρ – параметри алгоритму, які відображають ступінь значущості феромонного сліду та евристичної інформації ($q, \rho > 0, q + \rho = 1$).

q – величина, яка визначає «жадібність» алгоритму,

ρ – величина, яка визначає «стабільність» алгоритму,

R – кількість заявок.

Бажаність переходу визначимо наступним чином:

$$b_{ij} = \alpha(1 - v_{c_{1,x_{ij}}j}) + (1 - \alpha)(1 - w_{c_{2,x_{ij}}j}),$$

де $v_{c_{1,x_{ij}}j}$ – коефіцієнт схожості між учнем $c_{1,x_{ij}}$ та j -м ментором,

$w_{c_{2,x_{ij}}j}$ – коефіцієнт важливості $c_{2,x_{ij}}$ -ї предметної області для j -го ментора.

Критерій завершення. За критерій завершення візьмемо перевищення заданої кількості операцій.

Формалізуємо алгоритм ОМК:

Крок 1. Обираємо одну із наявних вершин-заявок випадковим чином.

Крок 1.1. Шукаємо допустимих менторів для даної заявки.

Крок 1.2. Для допустимих менторів знаходимо бажаність переходу та ймовірність переходу.

Крок 1.3. Обираємо найбільш ймовірний перехід і записуємо пару ментор-заявка у розклад.

Крок 2. Переходимо до наступної вершини-заявки.

Візьмемо $\lambda = 0.9$, $q = 0.6$, $\rho = 0.4$. Визначення, наскільки ці параметри впливають на мій алгоритм і наскільки вони оптимальні, — це предмет подальших досліджень.

Висновки

Підсистема «Інформаційна рекомендаційна система в сфері освітніх послуг» показуватиме розклад, який зможуть створювати та модифікувати ментори та учні прямо в даній системі, щоб ефективно розподіляти свій час. Використовуючи систему, учень зареєструється і створює заявку, в якій вказує, яким предметом і о котрій годині бажає мати заняття, а у менторів при цьому доступний розклад власних занять (календар, де видно вільні ментори чи ні). Програмно-алгоритмічне забезпечення системи повинно забезпечити

розв'язування задачі, яка полягає в тому, щоб знайти за певними критеріями найбільш відповідного ментора-однодумця, який зможе провести заняття в певний проміжок часу.

Зроблена математична постановка задачі та на основі неї прийнято рішення застосувати алгоритм локального пошуку як відправну точку і дослідженні даного типу задач. Виведено два правила, якими можливо користуватися, для реалізації евристичних алгоритмів для даної постановки задач. Розроблена процедура генерації сусідніх точок, використовуючи виведені правила. Формалізовано та розроблено алгоритми ЛП та ОМК.

3 РЕЗУЛЬТАТИ ЕКСПЕРИМЕНТАЛЬНИХ ДОСЛІДЖЕНЬ

Запропоновані алгоритми реалізовано на мові C#. Для оцінювання їх ефективності проведено обчислювальні експерименти на ПК з наступними характеристиками:

- процесор: AMD A10-5750M APU with Radeon™ HD Graphics 2.5 ГГц;
- об'єм оперативної пам'яті: 8 ГБ.

Порівнюємо час виконання алгоритмів (рис. 3.1). Критерієм завершення обох алгоритмів було обрано перевищення заданої операції. Тож час замірявся для 100 ітерацій. Встановимо параметр цільової функції $\alpha = 0.5$, оскільки будемо вважати, що для нас так само важлива схожість учня та ментора, як і збіг по предметним областям.



Рисунок 3.1 – Графік залежності часу виконання алгоритмів (100 ітерацій) від об'єму вхідних даних

Як бачимо з рисунка 3.1, що 1 ітерація алгоритму ОМК займає більше час ніж ітерація локального пошуку, і час ОМК на великих об'ємах даних збільшується більш швидко, ніж час виконання локального пошуку.

Давайте порівняймо найкращі значення цільових функцій за 100 ітерацій (рис. 3.2).

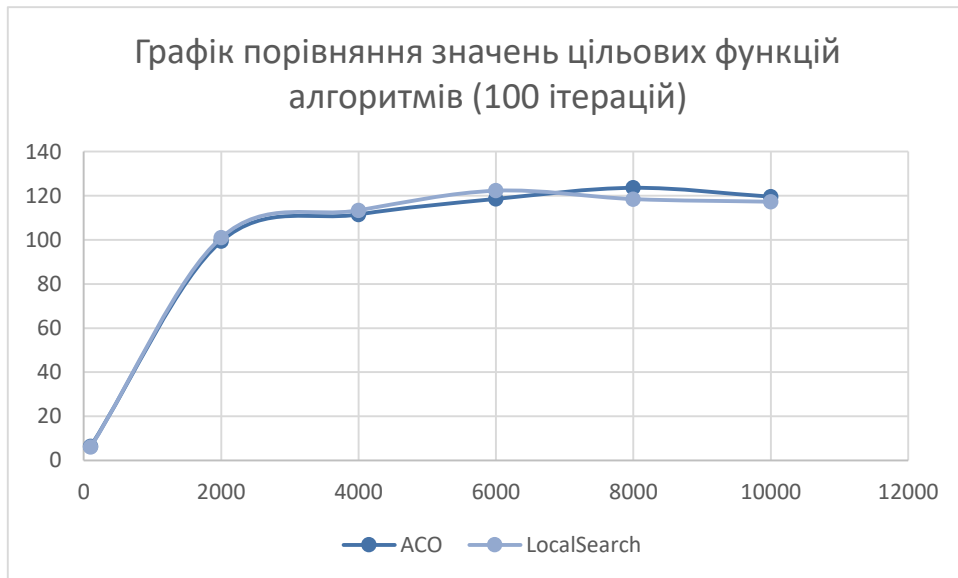


Рисунок 3.2 – Графік порівняння значень цільових функцій алгоритмів (100 ітерацій)

Як бачимо алгоритми мають близькі значення цільових функцій.

Порівняємо значення цільових функцій для 50 і 10 ітерацій (рис. 3.3,3.4).

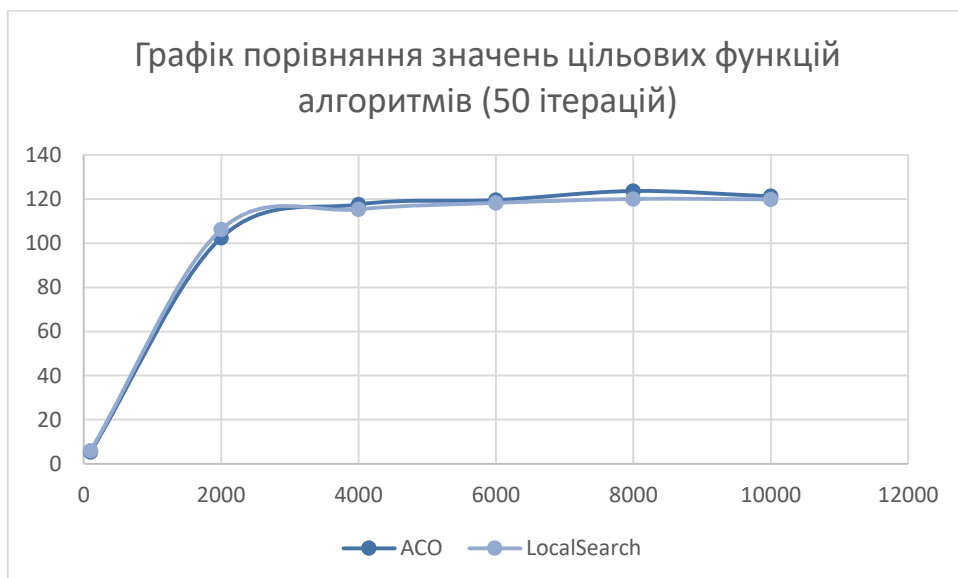


Рисунок 3.3 – Графік порівняння значень цільових функцій алгоритмів (100 ітерацій)

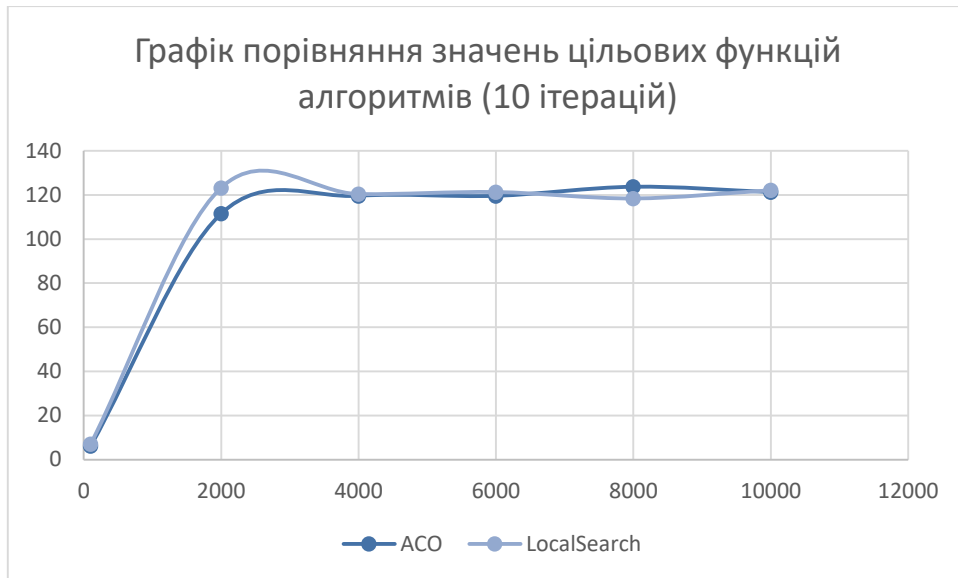


Рисунок 3.4 – Графік порівняння значень цільових функцій алгоритмів (10 ітерацій)

З рисунків 3.3 та 3.4 бачимо, що значення цільової функції для 50 ітерацій і 10 ітерацій близькі за значенням для обох алгоритмів.

Висновки. Отже, можемо зробити висновок, що локальний пошук ефективніший на першій стадії обчислень, бо займає менше часу. Проте перевагами ОМК, як впливає із проведених фахівцями досліджень, є можливість налаштування параметрів, що дозволяє знаходити покращені значення цільової функції за наявності достатнього часу для обчислень.

4 ОПИС ПРОГРАМНОГО ТА ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Архітектура програмного забезпечення

Архітектура ПЗ – це артефакт, що представляє собою результат процесу розробки програмного забезпечення. Елементи архітектури ПЗ і моделі їх з'єднання призначені для задоволення вимог до проєктованих системам. У проєкті архітектури ПЗ повинні бути враховані функціональні та нефункціональні вимоги до ефективності, витривалості, розширюваності, відмовостійкості, продуктивності, можливості повторного використання, а також адаптування розробляється ПО. [18]

Зобразимо як виглядає архітектура нашого програмного забезпечення (рис. 4.1).

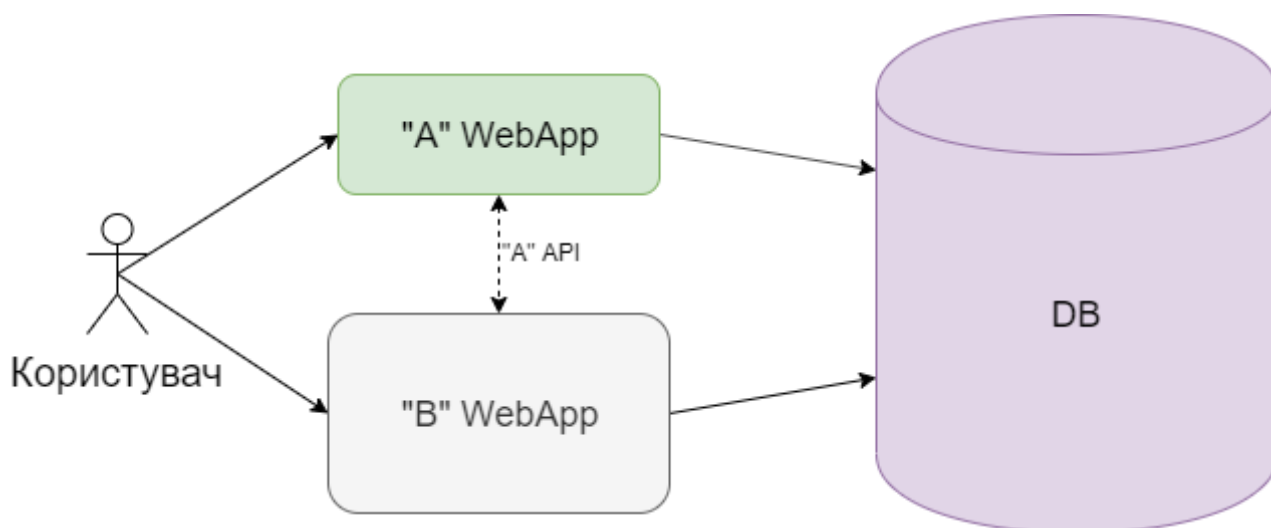


Рисунок 4.1 – Загальна схема взаємодії даної застосунку “А” з іншими елементами програмного забезпечення

З рисунку 4.1 видно, що є користувач, якому доступно два веб-застосунки “А” та “В”. “А” застосунок – це наша підсистема «Інформаційна рекомендаційна система в сфері освітніх послуг», яка може виступати як самостійний застосунок так і бути інтегрованою частиною іншої системи (системи “В”). Система “В” – система «Комплекс задач для підтримки комунікації менторів та учнів», у якій користувач має функціонал, як от можливість спілкуватися учням та менторам

через чат в режимі онлайн, пошук менторів з предметної області, пошук менторів по завданню чи проблемі, яку учень не може вирішити або зрозуміти, система оцінювання роботи менторів з боку учнів, а також надана можливість менторам публікувати статті з різних проблемних питань та областей, можливість знайти однодумців на основі власних уподобань, залучати нових знайомих і спілкуватися з ними.

Підсистема «Інформаційна рекомендаційна система в сфері освітніх послуг» є повністю автономною, має можливість бути інтегрованою, дякуючи реалізованому програмному інтерфейсу застосунку (Application Programming Interface – API). Підсистема «Інформаційна рекомендаційна система в сфері освітніх послуг» наразі має спільну базу з системою «Комплекс задач для підтримки комунікації менторів та учнів» і використовує дані, отримані через цю систему, які введені користувачами або проаналізовані і згруповані системою, наприклад, ментори-однодумці.

Підсистема «Інформаційна рекомендаційна система в сфері освітніх послуг» має функціонал, як от перегляд і модифікація розкладу для менторів та учнів, складання розкладу та сповіщення учнів та менторів про нові рекомендації для них, опираючись на побажання учнів та зайнятість менторів.

Отже, детальніше розглянемо підсистему «Інформаційна рекомендаційна система в сфері освітніх послуг». Програмне забезпечення цієї системи складається з серверної та клієнтської частин і його можна представити у вигляді схеми структурної розгортання як на рисунку 4.2.

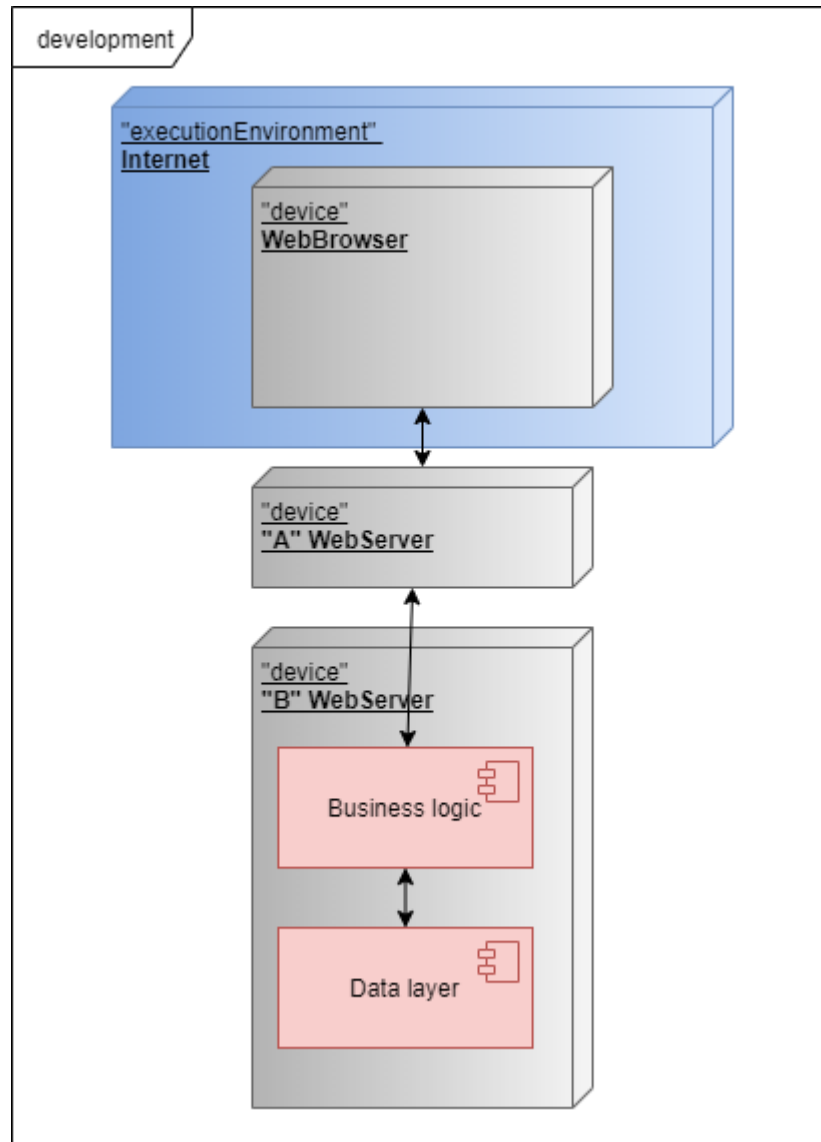


Рисунок 4.2 – Схема структурна розгортання

ПЗ має трьохрівневу архітектуру, що представлена на схемі структурній компонентів (рисунок 4.3) і складається з таких рівнів:

- Рівень 1. Користувацький інтерфейс.
- Рівень 2. Бізнес-логіка.
- Рівень 3. Дані.

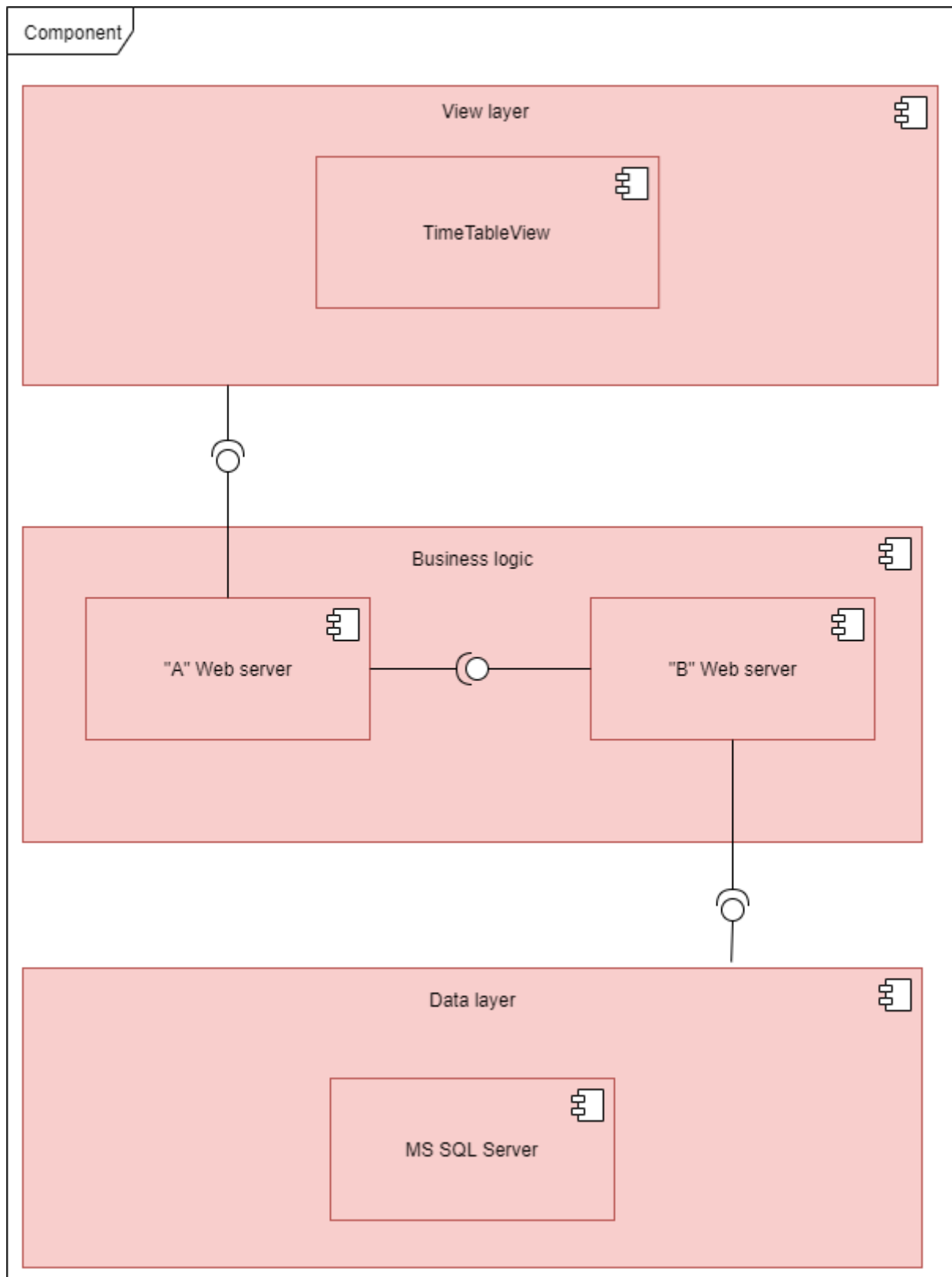


Рисунок 4.3 – Схема структурна компонентів

Кожен з рівнів має відповідний елемент програмного забезпечення:

- рівень 1 – браузер;

- рівень 2 – веб-застосунок;
- рівень 3 – база даних.

Отже, оскільки база даних в нас вже існує, в цьому розділі наведемо клієнтську і серверну частини ПЗ, що відповідають компонентам, що будуть взаємодіяти безпосередньо з браузером та веб-застосунком відповідно.

4.2 Засоби розробки

4.2.1 Серверна частина

Для створення серверної частини було обрано засоби, наведені нижче.

Фреймворк: ASP.NET Core.

Платформа ASP.NET Core являє собою технологію від компанії Microsoft, призначену для створення різного роду веб-додатків: від невеликих веб-сайтів до великих веб-порталів і веб-сервісів. [19]

З одного боку, ASP.NET Core є продовженням розвитку платформи ASP.NET. Але з іншого боку, це не просто черговий реліз. Вихід ASP.NET Core фактично означає революцію всієї платформи, її якісна зміна. [19]

ASP.NET Core тепер повністю є opensource-фреймворком. Всі вихідні файли фреймворку доступні на GitHub. [19]

ASP.NET Core може працювати над крос-платформним середовищем .NET Core, яке може бути розгорнуте на основних популярних операційних системах: Windows, Mac OS, Linux. І таким чином, за допомогою ASP.NET Core ми можемо створювати крос-платформні додатки. І хоча Windows, як середовище для розробки і розгортання програми досі переважає на ринку, але тепер вже ми не обмежені тільки цією операційною системою. Тобто з цим фреймворком ми зможемо запустити наш веб-застосунок не тільки на ОС Windows, але і на Linux і Mac OS. А для розгортання веб-застосунків можна використовувати традиційний IIS, або крос-платформний веб-сервер Kestrel. [19]

Завдяки модульності фреймворка всі необхідні компоненти веб-застосунків можуть завантажуватися як окремі модулі через пакетний менеджер

Nuget. Крім того, на відміну від попередніх версій платформи немає необхідності використовувати бібліотеку System.Web.dll. [19]

ASP.NET Core включає в себе фреймворк MVC, який об'єднує функціональність MVC, Web API і Web Pages. У попередніх версії платформи дані технології реалізувалися окремо і тому містили багато дублюючої функціональності. Зараз же вони об'єднані в одну програмну модель ASP.NET Core MVC. А Web Forms повністю пішли в минуле. [19]

Крім об'єднання вищезазначених технологій в одну модель в MVC був доданий ряд додаткових функцій. Однією з таких функцій є тег-хелпери (tag helper), які дозволяють більш органічно поєднувати синтаксис html з кодом C#. [20]

ASP.NET Core характеризується розширюваністю. Фреймворк побудований з набору щодо незалежних компонентів. І ми можемо або використовувати вбудовану реалізацію цих компонентів, або розширити їх за допомогою механізму спадкування, або зовсім створити і застосовувати свої компоненти зі своїм функціоналом. [19]

Також було спрощено управління залежностями і конфігурація проекту. Фреймворк тепер має свій легкий контейнер для впровадження залежностей, і більше немає необхідності застосовувати сторонні контейнери, такі як Autofac, Ninject. Хоча при бажанні їх також можна продовжувати використовувати. [19]

В якості інструментарію розробки ми можемо використовувати останні випуски Visual Studio, починаючи з версії Visual Studio 2015. Крім того, ми можемо створювати додатки в середовищі Visual Studio Code, яка є крос-платформної і може працювати як на Windows, так і на Mac OS X і Linux. [19]

Для обробки запитів тепер використовується новий конвеєр HTTP, який заснований на компонентах Katana і специфікації OWIN. А його модульність дозволяє легко додати свої власні компоненти. [19]

Якщо підсумувати, то можна виділити наступні ключові переваги ASP.NET Core:

- новий легкий і модульний конвеєр HTTP-запитів;
- можливість розгортати додаток як на IIS, так і в рамках свого власного процесу;
- використання платформи .NET Core і її функціональності;
- поширення пакетів платформи через NuGet;
- інтегрована підтримка для створення та використання пакетів NuGet;
- єдиний стек веб-розробки, що поєднує Web UI і Web API;
- конфігурація для спрощеного використання в хмарі;
- вбудована підтримка для впровадження залежностей;
- можливість розширення;
- крос-платформність: можливість розробки і розгортання додатків ASP.NET на Windows, Mac і Linux;
- розвиток як open source, відкритість до змін. [19]

Ці та інші особливості і можливості стали основою для нової моделі програмування. [19]

Для розробки програмного забезпечення з усього перерахованого в цьому фреймворку я використаю ASP.NET Core Web API, що чудово вписується в нашу архітектуру, легко поєднується з іншими елементами програмного забезпечення і буде крос-платформним веб-застосунком, що дозволить в майбутньому розгорнути його на будь-якій платформі, в тому числі в хмарі і робити розрахунки і складання розкладу в хмарному середовищі.

Мова програмування: C#.

Основною мовою програмування технології ASP.NET Core є мова C#. У цій технології єдиний спосіб опису програмного коду (метадані), одне середовище виконання (Common Language Runtime) і одна базова бібліотека (BCL). Міжмовна інтеграція досягається за рахунок того, що програма, написана на мові високого рівня (C# чи Basic, чи C++, чи ін.) не одразу компілюється у машинний код, а спочатку відбувається компіляція у команди (псевдокод) єдиної

проміжкової мови Intermediate Language (IL). Незалежність від платформи досягається за рахунок того, що під час виконання програми команди проміжкової мови перетворюються у машинні команди у середовищі Common Language Runtime (CLR) за допомогою відповідних компіляторів (JIT-компіляторів). [20]

Середовище розробки: Visual Studio 2019.

Microsoft Visual Studio — серія продуктів фірми Майкрософт, які включають інтегроване середовище розробки програмного забезпечення та ряд інших інструментальних засобів. Ці продукти дозволяють розробляти як консольні програми, так і програми з графічним інтерфейсом, в тому числі з підтримкою технології Windows Forms, а також веб-сайти, веб-застосунки, веб-служби як в рідному, так і в керованому кодах для всіх платформ, що підтримуються Microsoft Windows, Windows Mobile, Windows Phone, Windows CE, .NET Framework, .NET Compact Framework та Microsoft Silverlight. [21]

Visual Studio (VS) складається з редактора коду, що підтримує IntelliSense і рефакторинг коду. Вбудовану панель можна використовувати як для вирівнювання вихідного коду, так і як клавіатурного шаблону. Інші вбудовані інструменти включають [21]:

- редактор форм для розробки додатків GUI;
- веб-редактор;
- редактор класів;
- редактор схем баз даних.

VS дає можливість встановити плагіни для розширення функціональних можливостей майже на будь-якому рівні, в тому числі для підтримки систем контролю вихідного коду (таких як Subversion та GitHub) та нових інструментів для редагування на предметно-орієнтованих мовах програмування, а також інструментів для інших аспектів життєвого циклу розробки програмного забезпечення (таких як клієнт Team Foundation Server, TFS: Team Explorer). [21]

Visual Studio надає мовні сервіси, які дозволяють коду редактора та debugger (debugger) підтримувати практично будь-яку мову програмування. [21]

Так само Visual Studio підтримує XML / XSLT, HTML / XHTML, JavaScript та CSS. [21]

СКБД: SQL Server 2017.

Microsoft SQL Server 2017 – комерційна система керування базами даних, що розповсюджується корпорацією Microsoft. Мова, що використовується для запитів – Transact-SQL, створена спільно Microsoft та Sybase. Transact-SQL є реалізацією стандарту ANSI/ISO щодо структурованої мови запитів SQL із розширеннями. Використовується як для невеликих і середніх за розміром баз даних, так і для великих баз даних масштабу підприємства. Багато років вдало конкурує з іншими системами керування базами даних. [22]

Зазначу, чому саме обрала цю СКБД.

Ви самі вибираєте мови та платформи. Ви вибираєте мову, який вам підходить, як на локальній платформі, так і в облаках - тепер підтримують Windows, Linux і контейнери Docker. [22]

Виробництво на рівні лідерів галузі. Можливість масштабування, що дозволяє досягти високої продуктивності і доступності для критично важливих інтелектуальних застосунків та сховищ даних. [22]

Найменш вразлива база даних. Забезпечується захист даних як під час зберігання, так і передачі, використовуючи базу даних, яка вже сім років визнається найменш вразливою за результатами тестів на вразливість в Національному інституті стандартів і технологій США (NIST). [22]

Аналітика в режимі реального часу. Можливість аналітика в реальному часі, яка надає до мільйона прогнозувань в секунду. [22]

Комплексна мобільна бізнес-аналітика. Перетворення необроблених даних у звіти, доступних на будь-яких пристроях (включаючи пристрої під управлінням Windows, Android та iOS). [22]

Витримує високе навантаження. SQL Server використовує у всіх тестах на навантаження робочі навантажувачі TPC-E OLTP та навантажувачі TPC-H зі збереженням даних. [22]

Висока продуктивність. Можливість використання різних варіантів – від хмарних даних до великих корпоративних сховищ, можливість скоротити вимоги до вільного місця, завдяки вдосконаленому алгоритму роботи даних. Можливість масштабування петабази даних, використовуючи збережені реляційні дані корпоративного рівня, а також можливість інтеграції нереляційних джерел, такі як Hadoop. [22]

Безпечність і відповідність вимог. Захищені дані про час користування та передачі з підтримкою бази даних. Нововведення SQL Server 2017 у безпеці захищають дані для критично важливих робочих навантажувачів, які підтримують безпечний режим роботи, і забезпечують технологією постійного шифрування на рівні передачі даних, прозорі шифрування даних (TDE), динамічне маскування даних та надійний аудит. [22]

Висока доступність і аварійне навчання. Забезпечує більше часу безперервної роботи з критично важливими активами, завдяки цьому можна буде працювати з робочими повідомленнями, залученими з Інтернету, і балансувати дані, доступні для використання в базі даних для початкового користування. Крім того, можна помістити асинхронну репліку у віртуальній машині Azure для забезпечення високої продуктивності для гібридного рішення. [22]

Використання будь-яких обсягів даних. Можливість поєднання реляційних даних та великих даних з технологією PolyBase, яка дозволяє виконати запит Hadoop за допомогою простої команди T-SQL. Підтримка JSON дозволяє легко проаналізувати та зберегти документи JSON та виводити реляційні дані у файли JSON. У SQL Server 2017 тепер можна керувати графовими даними та виконувати запити для них всередині реляційної бази даних. [22].

Провідна аналітика в базі даних. Можливість побудови інтелектуальних звітів із застосуванням R та Python з підтримкою сервісного обладнання

машинного навчання SQL Server. Переходите від простого реагування на аналіз прогнозу та прогнозування аналітики, виконуючи провідну аналітику, безпосередньо в базі даних. Застосовуючи багатопоточну обробку з масовим паралелізмом, ви отримаєте отримані цінні відомості, що використовуються лише R з відкритим вихідним кодом і Python. [22].

Прості у використанні інструменти та конектори. Можливість використання накопиченого досвіду та звичних засобів, такі як Azure Active Directory та SQL Server Management Studio, щоб управляти інфраструктурою базових даних у локальній мережі SQL Server та Microsoft Azure. Можливість застосувати стандартні API для різних платформ і оновленні засоби розробки з Visual Studio, що дозволить створити додатки нового покоління для Інтернету, корпоративних середніх, бізнес-аналітиків і мобільних пристроїв. [22]

Шаблони проектування: Repository, UnitOfWork.

Repository. Одним з найбільш часто використовуваних паттернів при роботі з даними є шаблон *Repository*. Репозиторій може абстрагуватися від конкретних підключень до джерела даних, з якими працює програма, і він є проміжною ланкою між класами, безпосередньо взаємодіючими з даними і з рештою програми. [23]

UnitOfWork. Обслуговує набір об'єктів, змінюваних в бізнес-транзакції (бізнес-дії) і управляє записом змін і вирішенням проблем конкуренції даних. Коли необхідно писати і читати дані з БД, важливо стежити за тим, що ви змінили і якщо не змінили - не записати дані в БД. Також необхідно вставляти дані про нові об'єкти і видаляти дані про старі. Можна записувати в БД кожен змінюваний об'єкт, але це призведе до великої кількості дрібних запитів до БД, що закінчиться уповільненням роботи програми. Більш того, це вимагає тримати відкриту транзакцію весь час роботи програми, що непрактично, якщо додаток обробляє кілька запитів одночасно. Ситуація ще гірша, якщо необхідно стежити за читанням з БД, щоб уникнути неконсистентного читання. Реалізація патерну *UnitOfWork* стежить за всіма діями додатку, які можуть змінити БД в рамках

одного бізнес-дії. Коли бізнес-дія завершується, UnitOfWork виявляє всі зміни і вносить їх в БД. [24]

Робота з базами даних засобами LINQ.

Microsoft Language Integrated Query (LINQ) пропонує розробникам новий спосіб здійснення запитів даних з використанням строго типізованих запитів і строго типізованих результатів, загальних за цілою низкою різноманітних типів даних, включаючи реляційні бази даних, об'єкти .NET і XML. При використанні строго типізованих запитів і результатів, LINQ підвищує продуктивність праці розробників з перевагами IntelliSense і перевірки помилок під час компіляції [25].

LINQ to SQL – це фреймворк об'єктно-реляційного відображення (ORM), який впроваджує пряме, "один до одного", відображення бази даних Microsoft SQL Server для класів .NET, а також запити результуючих об'єктів з використанням LINQ [25].

Основний сценарій використання LINQ to SQL — це розробка додатків із швидким циклом розробки і простим об'єктом із реляційним відображенням "один до одного" поряд із базою даних сімейства Microsoft SQL Server. Іншими словами, при створенні програми, чия об'єктна модель будується дуже схожою на існуючу структуру бази даних, або коли база даних для програми ще не існує, і немає схильності до створення схеми бази даних, яка відображає об'єктну модель; ви можете використовувати LINQ to SQL, щоб відобразити підмножину таблиць безпосередньо до класів з необхідними стовпцями з кожної таблиці, представленої як властивості відповідного класу [25].

4.2.2 Клієнтська частина

Для створення клієнтської частини було обрано наступні засоби:

Фреймворк: Angular.

Angular представляє фреймворк від компанії Google для створення клієнтських додатків. Перш за все він націлений на розробку SPA-рішень (Single Page Application), тобто односторінкових застосунків. В цьому плані Angular є

спадкоємцем іншого фреймворку AngularJS. Водночас Angular це не нова версія AngularJS, а принципово новий фреймворк. [26]

Angular надає таку функціональність, як двостороннє зв'язування, що дозволяє динамічно змінювати дані в одному місці інтерфейсу при зміні даних моделі в іншому, шаблони, маршрутизацію і так далі. [26]

Однією з ключових особливостей Angular є те, що він використовує як мову програмування мову TypeScript. Але ми не обмежені мовою TypeScript. Якщо є бажання, можемо писати програми на Angular за допомогою таких мов, як Dart або JavaScript. [26]

Однак, оскільки TypeScript все-таки є основною мовою для Angular, я обрала її, як основну мову програмування для клієнтської частини.

Мова програмування: TypeScript.

TypeScript — мова програмування, що позиціюється як засіб розробки веб-застосунків, що розширює можливості JavaScript. [26]

Код експериментального компілятора, котрий трансліює код TypeScript в представлення JavaScript, поширюється під ліцензією Apache, розробка ведеться в публічному репозитарії через сервіс CodePlex. Специфікації мови відкриті і опубліковані в рамках угоди Open Web Foundation Specification Agreement (OWFa 1.0). [26]

TypeScript є зворотно сумісною з JavaScript. Фактично, після компіляції програму на TypeScript можна виконувати в будь-якому сучасному браузері або використовувати спільно з серверною платформою Node.js. [26]

Переваги TypeScript над JavaScript:

- можливість явного визначення типів (статична типізація),
- підтримка використання повноцінних класів (як в традиційних об'єктно-орієнтованих мовах),
- підтримка підключення модулів. [26]

Node.js — це JavaScript-оточення побудоване на JavaScript-рушієві Chrome V8. Якщо раніше Javascript застосовувався для обробки даних в браузері

користувача, то node.js надав можливість виконувати JavaScript-скрипти на сервері та відправляти користувачеві результат їх виконання. Платформа Node.js перетворила JavaScript на мову загального використання з великою спільнотою розробників. [27]

Node.js має наступні властивості [27]:

- асинхронна одно-нитева модель виконання запитів;
- неблокуючий ввід/вивід;
- система модулів CommonJS;
- рушій JavaScript Google V8.

Bootstrap — це безкоштовний набір інструментів з відкритим кодом, призначений для створення веб-сайтів та веб-додатків, який містить шаблони CSS та HTML для типографіки, форм, кнопок, навігації та інших компонентів інтерфейсу, а також додаткові розширення JavaScript. Він спрощує розробку динамічних веб-сайтів і веб-додатків. Bootstrap – це клієнтський фреймворк, тобто інтерфейс для користувача, на відміну від коду серверної сторони, який знаходиться на сервері [28].

FontAwesome – це набір інструментів для шрифтів і значків на основі CSS та LESS. Це було зроблено Дейвом Генді для використання з Twitter Bootstrap, а пізніше було включено до BootstrapCDN. Шрифт Awesome має частку ринку на 20% серед тих веб-сайтів, які використовують незалежні шрифтові скрипти на своїй платформі, зайнявши друге місце після шрифтів Google [29].

4.3 Вимоги до технічного забезпечення

Для правильної роботи розробленої системи необхідно правильно розвернути всі її компоненти. Клієнтська частина не потребує розгортання. Проте для роботи з даною системою є все ж певні вимоги, а саме:

- браузер Google Chrome 60 і вище;
- наявність інтернет-з'єднання.

На серверній стороні необхідно розвертати програмне забезпечення, маючи при цьому всі вихідні файли програми і необхідні бібліотеки скомпільовані під обрану операційну систему. Технічні засоби для розгортання серверної частини включають:

а) комп'ютер:

- 1) процесор з частотою не менше 2 ГГц;
- 2) об'єм оперативної пам'яті не менше 8 ГБ.

б) програмне забезпечення:

- 1) операційна система Windows 10 і вище або Ubuntu;
- 3) встановлено SQL Server 2017 і вище;
- 4) сервер застосувань – IIS7 і вище (для Windows 10);
- 5) система управління базами даних SQL Management Studio (версія повинна бути сумісною з версією SQL Server).

в) комп'ютерна периферія:

- 1) монітор;
- 2) мишка;
- 3) клавіатура;

г) наявність інтернет-з'єднання.

5 РОЗРОБКА СТАРТАП-ПРОЕКТУ

Отримавши і проаналізувавши результати експериментальних досліджень дійшли висновку, що маємо вагомі наукові досягнення і є над чим продовжувати працювати. Мурашиний алгоритм можна і потрібно застосовувати для знаходження розкладу для даної задачі. Було реалізоване програмне забезпечення, що описано у розділі 4, та застосовано розроблений мурашиний алгоритм. Доцільним є проаналізувати розроблений програмний продукт на те, наскільки він є конкурентоспроможним на ринку, чи можна на основі цього продукту запустити стартап-проект і чи буде такий проект рентабельним. Це і буде зроблено в даному розділі.

У кожного стартап-проекту є назва. У цьому розділі розроблений програмний продукт будемо називати «EasyHelp», основне призначення якого – підтримка комунікації менторів та учнів та інтелектуалізація процесів онлайн-освіти.

Метою розробки є підвищення ефективності процесу навчання за рахунок полегшення пошуку менторів та учнів та організації пошуку однодумців для спілкування; підвищення якості інформування потенційних споживачів та інтелектуалізація процесів надання освітніх послуг онлайн.

Мій проект є стартапом, бо:

- немає аналогів в Україні - застосувань зі згаданим функціоналом ще ніде немає, всі застосування охоплюють лише маленьку частинку, того що стосується пошуку менторів;
- є новизна – пошук менторів-однодумців та рекомендація розкладу за допомогою розроблених алгоритмів є унікальною технологією;
- на розкрутку проекту потрібні гроші.

5.1 Опис ідеї проекту

У цьому пункті опишемо зміст ідеї, що пропонується, та можливі напрямки застосування, основні вигоди, що може отримати користувач програмного

продукту (табл. 5.1) та чим відрізняється від існуючих аналогів та замінників (табл. 5.2). Наведемо таблицю (табл. 5.1), що дасть цілісне уявлення про зміст ідеї та можливі базові потенційні ринки, в межах яких потрібно шукати групи потенційних клієнтів.

Таблиця 5.1 – Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Рекомендаційна система для онлайн освіти, в якій учні та ментори можуть ефективно взаємодіяти один з одним	1. Пошук менторів, що можуть бути потенційними однодумцями	Користувач (учень) не просто знаходити ментора, який може проконсультувати його в деяких питаннях, а й має схожі погляди на життя на спільні інтереси, що полегшує взаємодію та спілкування
	2. Кабінет ментора	Ментор може тримати всю інформацію в одному місці, наприклад, розклад, учні, з якими проводиться навчання, потенційні учні
	3. Заявки учнів	Можливість поставити проблему у вигляді заявки, і не чекати відповідь як на форумі, а отримати консультацію від ментора онлайн
	4. Розклад онлайн	Можливість менторам вести свій розклад, учням – подавати заявки на певний час заняття, при цьому система рекомендує ментора, який вільний у бажаний час і найбільше підходить за кваліфікацією і уподобаннями
	5. Чат для спілкування	Можливість спілкуватися один з одним не використовуючи сторонні ресурси

Аналіз потенційних техніко-економічних переваг ідеї (чим відрізняється від існуючих аналогів та замінників) порівняно із пропозиціями конкурентів включає:

а) визначення переліку техніко-економічних властивостей та характеристик ідеї;

б) визначення попереднього кола конкурентів (проектів-конкурентів), що вже існують на ринку, та проводиться збір інформації щодо значень техніко-економічних показників для ідеї власного проекту та проектів-конкурентів відповідно до визначеного вище переліку;

в) порівняльний аналіз показників: для власної ідеї визначаються показники, що мають:

- 1) гірші значення (W, слабкі);
- 2) аналогічні (N, нейтральні) значення;
- 3) кращі значення (S, сильні) (табл. 5.2).

Таблиця 5.2 – Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№ п/п	Техніко-економічні характеристики ідеї	Товари/концепції конкурентів (потенційні)				W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		Мій проект	Конкурент1 – сайт «Ваш репетитор»	Конкурент2 – сайт «Preply»	Конкурент3 – сайт «Buki»			
1	Рівень автоматизації	У процесах не беруть участь адміністратори і модератори сайту	Адміністратори сайту самі вирішують кому надіслати заявку і повністю контролюють процес занять менторів та учнів	Адміністратори зв'язуються з менторами та учнями	У процесах не беруть участь адміністратори і модератори сайту			+

Продовження таблиці 5.2

№ п/ п	Техніко- економічні характерист ики ідеї	Товари/концепції конкурентів (потенційні)				W (сла бка сто рон а)	N (не йт ра ль на сто ро на)	S (сил ьна стор она)
		Мій проект	Конкуре нт1 – сайт «Ваш репетит ор»	Конкур ент2 – сайт «Preply »	Конкуре нт3 – сайт «Buki»			
2	Інтерфейс	Зручний, сучасний і привабливи й інтерфейс	Старий інтерфейс, без можливо сті користув атися нормаль ним чатом	Інтерфе йс кращий ніж у першого конкуре нта, але залишає бажати кращого , інтерфе йс незручн ий	Інтерфейс кращий ніж у першого конкурен та, але залишає бажати кращого			+
3	Кількість функціоналу доступного користувачу	Заявки, Статті, Введення кабінетів учня та ментора, Чат (міні та великий), Введення розкладу, Пошук менторів- однодумців, Анкетуванн я	Заявка, примітив не спілкува ння з адмініст ратором та заповнен ня повідомл ень про проведен е заняття	Пошук репетит орів, спілкува ння з адмініст ратором	Пошук менторів , чат з менторо м.			+

Продовження таблиці 5.2

№ п/ п	Техніко- економічні характерист ики ідеї	Товари/концепції конкурентів (потенційні)				W (сла бка сто рон а)	N (не йт ра ль на сто ро на)	S (сил ьна стор она)
		Мій проект	Конкуре нт1 – сайт «Ваш репетит ор»	Конкур ент2 – сайт «Preply »	Конкуре нт3 – сайт «Buki»			
4	Вартість користуванн я	Безкоштовн о	Знімаєть ся відсоток з проведен их занять	Безкошт овно	Безкошт овно		+	
5	Направлення на вирішення специфічних задач	Широко направлени й	Вузько направле ний	Вузько направл ений	Вузько направле ний			+

Отже, на основі таблиці 5.2 можемо бачити, що проект конкурентоспроможний.

5.2 Технологічний аудит ідеї проекту

В межах даного підрозділу проведемо аудит технології, за допомогою якої можна реалізувати ідею проекту (технології створення товару). Визначення технологічної здійсненності ідеї проекту передбачає аналіз таких складових (табл. 5.3):

- за якою технологією буде виготовлено товар згідно ідеї проекту;
- чи існують такі технології, чи їх потрібно розробити/доробити;
- чи доступні такі технології авторам проекту.

Таблиця 5.3 – Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології реалізації	Наявність технологій	Доступність технологій
1	Пошук менторів, що можуть бути потенційними	Методи кластеризації, реалізовані на мові C#	Наявні	Розроблені у моїй бакалаврській роботі
2	однодумцями	Локально-чутливе хешування, реалізоване на мові Java	Наявні	Розроблено у бакалаврській роботі однокурсника
3	Кабінет ментора та	Framework ASP.NET MVC	Наявні	Доступні
4	заявки учнів	Framework ASP.NET Core	Наявні	Доступні
5	Розклад онлайн	Мурашині алгоритми	Опис технології наявний, але під конкретну задачу треба розробляти	Опис алгоритмів в загальному доступний, можливість розробити ці алгоритми є
Обрана технологія реалізації ідеї проекту: Framework ASP.NET Core, Мурашині алгоритми, Методи кластеризації.				

Отже, бачимо, що всі описані технології нам доступні, деякі з них (мурашині алгоритми та методи кластеризації) треба реалізувати конкретно під наш проект. Обрані технології Framework ASP.NET Core, оскільки на сьогодні

це перспективна і новітня платформа для розробки Web-застосунків, а також обрано методи кластеризації так як вони вже реалізовані конкретно під на проект і доступні.

5.3 Аналіз ринкових можливостей запуску стартап-проекту

У цьому пункті визначимо ринкові можливості, які можна використати під час ринкового впровадження проекту, та ринкових загроз, які можуть перешкодити реалізації проекту, що дозволить спланувати напрями розвитку проекту із урахуванням стану ринкового середовища, потреб потенційних клієнтів та пропозицій проектів-конкурентів.

Спочатку проводиться аналіз попиту: наявність попиту, обсяг, динаміка розвитку ринку (табл. 5.3).

Таблиця 5.3 – Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	5
2	Загальний обсяг продаж, грн/ум.од	24 000
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу (вказати характер обмежень)	Треба виділити великі гроші на рекламу
5	Специфічні вимоги до стандартизації та сертифікації	Немає
6	Середня норма рентабельності в галузі (або по ринку), %	120 %

Отже, можемо зробити висновок, що проект рентабельний і є сенс вкладати в нього гроші і продовжувати працювати над ним.

Надалі визначимо потенційні групи клієнтів, їх характеристики, та сформуємо орієнтовний перелік вимог до продукту для кожної групи (табл. 5.5).

Таблиця 5.5 – Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1	Знайти ментора під час навчання в університеті	Студенти	Шукаю менторів за малу ціну, часто непотрібні ментори на довгий строк, а просто щоб допомогли в конкретному питанні	Можливість створити запит (заявку) з вказанням конкретної проблематики і ментор, який відповість на цю заявку має бути компетентний у питанні, що вказав студент
2	Знайти ментора під час навчання в школі	Школярі	Шукають репетиторів на довгий строк	Можливість знайти ментора на довгий строк по предметній області
3	Пошук ментора під час перекваліфікації	Люди середнього віку з певним досвідом роботи в певній сфері	Зазвичай самостійно вивчають матеріал, та іноді стопоряться на якомусь питанні, для чого і шукають ментора	Можливість створити запит (заявку) з вказанням конкретної проблематики і ментор, який відповість на цю заявку має бути компетентний у питанні, що вказав студент
4	Мати всю інформацію в одному місці	Ментори	Вимога не лазити по сайтам і не створювати свій документ Word, щоб ввести всіх своїх учнів	Кабінет ментора
5	Бажання зростати як спеціаліст,	Працюючі люди з досвідом	Вимога мати можливість час від часу надавати	Рекомендаційна система

	допомагаючи іншим		послуги, кому це необхідно	
--	-------------------	--	----------------------------	--

Після визначення потенційних груп клієнтів проведемо аналіз ринкового середовища: складемо таблиці факторів, що сприяють ринковому впровадженню проекту, та факторів, що йому перешкоджають (табл. 5.6-5.7).

Таблиця 5.6 – Фактори загроз

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Збільшення конкуренції	Можливо з'являться багато конкурентів, і на ринку буде важко виділитися	Придумувати оригінальний функціонал і сильну рекламу
2	Швидкість розвитку технологій	Можливо за пару років з'являться такі технології, що дана система уже застаріє	Бути гнучкими і на льоту впроваджувати нові технології
3	Нестабільність ринку України	Війна, нестабільна політична ситуація, інфляція, низький рівень економіки	Вихід на європейський ринок

Таблиця 5.7 – Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Інвестиції	Зацікавленість інших осіб у даному стартапі	Виконати всі вимоги інвестори
2	Стати першопрохідцем з даною ідеєю	Хоч в Україні є аналоги, але функціонал цих аналогів бідний і немає ніяких складних обрахунків, це просто пошукові сайти. Є можливість стати не просто пошуковим сайтом а платформою, яка давати можливість мати кабінет, рекламувати себе, система сам буде пропонувати менторів-однодумців, що вам підходять, а вся інформація про заняття буде зберігатися в одному місці	Реалізувати весь функціонал найкращим чином і стати номер 1 на ринку

3	Розвивати дану ідею	Ідея немає обмежень і її можна і далі розвивати додаючи в систему і штучний інтелект та інші перспективні технології, що зацікавить і користувачів і інвесторів	Розвивати ідею далі
---	---------------------	---	---------------------

Проведемо аналіз пропозиції: визначимо загальні риси конкуренції на ринку (табл. 5.8).

Таблиця 5.8 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1. Тип конкуренції – Чиста	Всі конкуренти однотипні, ніхто не вирізняється	Внести новий функціонал і запропонувати свіже рішення, якого на ринку немає
2. За рівнем конкурентної боротьби – Національний	Всі продукти конкурентів доступні на всій території України	Охопити весь ринок України
3. За галузевою ознакою - Внутрішньогалузева	Галузь ІТ	Впровадження нових технологій. Намагатися вплинути на ще одну галузь – освіта (в т. ч. онлайн-освіта)
4. Конкуренція за видами товарів - Між бажаннями	Задовольняють одне бажання – знайти ментора	Аналізувати (автоматизовано) бажання клієнтів і впроваджувати зміни на основі зміни
5. За характером конкурентних переваг – Нецінова	Оскільки сервіси безкоштовні йде боротьба за допомогою зручності та	Підтримувати сучасний та зручний інтерфейс та

	сучасності інтерфейсу і кількості функціоналу	впроваджувати новий функціонал
6. За інтенсивністю - Не марочна	Не на слуху у всіх	Стати брендом

Після аналізу конкуренції проведемо більш детальний аналіз умов конкуренції в галузі (за моделлю 5 сил М. Портера) (табл. 5.9). Визначимо, чи можливо працювати на ринку з огляду на конкурентну ситуацію, а також сильні сторони, які повинен мати проект, щоб бути конкурентоспроможним на ринку.

Таблиця 5.9 – Аналіз конкуренції в галузі за М. Портером

Складові і аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари замітники
	Сайт «Ваш репетитор» Сайт «Preply» Сайт «Buki»	Розмір капіталовкладень – потрібен великий	Немає	Клієнти змінюють бажання часто, треба гнучко реагувати	Ці сайти між собою взаємозамінні
Висновки:	Боротьба середньої інтенсивності, виграє той, хто вклав більше грошей в рекламу і краще пристосувався	Можна увійти на ринок, проте треба мати великий початковий капітал, зокрема для реклами	Не диктують, бо в даній галузі їх немає	Клієнти безумовно диктують умови роботи на ринку. Від задоволення побажань клієнтів щодо наявності і якості функціоналу	Ніяких обмежень на ринку не виникне, бо аналогів цьому сайту не буде

	я бо бажань клієнтів			буде залежати чи будуть вони користуватися вашою системою	
--	-------------------------	--	--	--	--

Отже, проект є конкурентоспроможним, незважаючи на існування віддалених аналогів. Сильними сторонами проекту є новий функціонал, високий рівень автоматизації (заявки учнів не оброблюють адміністратори чи модератори), зручний, привабливий та сучасний інтерфейс, якого немає у конкурентів.

На основі аналізу конкуренції, наведеного в таблиці 5.9, а також із урахуванням характеристик ідеї проекту (табл. 5.2), вимог споживачів до товару (табл. 5.5) та факторів маркетингового середовища (табл. 5.6-5.7) визначимо та обґрунтуємо перелік факторів конкурентоспроможності (табл. 5.10).

Таблиця 5.10 – Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1	Динаміка галузі	ІТ галузь дійсно динамічна, кожного дня з'являються нові технології і відбувається дуже швидкий розвиток конкурентів
2	Постійні витрати	На підтримку таких систем потрібні постійні витрати на сервера, впровадження нових рішень, підтримку програмних продуктів
3	Наявність торгівельних знаків	Мати унікальну і відому дуже важливо, на це клієнти звертають увагу
4	Рівень автоматизації	Сьогодні чим більший рівень автоматизації, тим менше затрат на підтримку та робота і

		компанії, і клієнтів налаштована ефективніше
5	Бар'єри проникнення	Потрібно ставати брендом №1 і впроваджувати такі технологічні інновації, щоб іншим було важко увійти в дану галузь

За визначеними факторами конкурентоспроможності (табл. 5.10) проведемо аналіз сильних та слабких сторін стартап-проекту (табл. 5.11).

Таблиця 5.11 – Порівняльний аналіз сильних та слабких сторін "EasyHelp"

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з "EasyHelp"						
			-3	-2	-1	0	1	2	3
1	Динаміка галузі	15	+						
2	Постійні витрати	20						+	
3	Наявність торговельних знаків	14					+		
4	Рівень автоматизації	18			+				
5	Бар'єри проникнення	9				+			

Фінальним етапом ринкового аналізу можливостей впровадження проекту є складання SWOT-аналізу (матриці аналізу сильних (Strength) та слабких (Weak) сторін, загроз (Troubles) та можливостей (Opportunities) (табл. 5.12) на основі виділених ринкових загроз та можливостей, та сильних і слабких сторін (табл. 5.11). Перелік ринкових загроз та ринкових можливостей складемо на основі аналізу факторів загроз та факторів можливостей маркетингового середовища.

Таблиця 5.12 – SWOT- аналіз стартап-проекту

Сильні сторони: до звичайно пошуку додається абсолютно новий функціонал, автоматизованість	Слабкі сторони: мало ресурсів та часу для створення дійсно якісного продукту
--	--

системи, все зберігається в одному місці, зручний інтерфейс	
Можливості: зацікавити інвесторів, розвивати продукт далі	Загрози: високий рівень капіталовкладень, нестабільність економічного і політичного стану в Україні

На основі SWOT-аналізу розробляються альтернативи ринкової поведінки (перелік заходів) для виведення стартап-проекту на ринок та орієнтовний оптимальний час їх ринкової реалізації з огляду на потенційні проекти конкурентів, що можуть бути виведені на. Визначені альтернативи аналізуються з точки зору строків та ймовірності отримання ресурсів (табл. 5.13).

Таблиця 5.13 – Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Реалізувати проект та шукати інвесторів	75%	1 рік
2	Шукати інвесторів, а потім реалізовувати проект	100%	2 роки
3	Брати позику в банку і реалізовувати проект	60%	1 рік
4	Накопичувати гроші на рекламу і реалізовувати проект	99%	3 роки

Оберемо найкращу альтернативу з точки зору найбільшої ймовірності та найбільш стислих строків – реалізувати проект та шукати інвесторів, адже брати

позику в банку дуже ризикова, і велика ймовірність, що її не дадуть, або потім не зможеш розплатитися.

Отже, спочатку реалізовуємо проект, щоб було, що презентувати, а потім шукаємо інвесторів.

5.4 Розроблення ринкової стратегії проекту

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: опис цільових груп потенційних споживачів (табл. 5.14).

Таблиця 5.1 – Вибір цільових груп потенційних споживачів

№ п/ п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті (від 1 до 10)	Простота входу у сегмент (від 1 до 10; 1 – дуже просто, 10 – дуже непросто)
1	Студенти	+	100%	9	1
2	Школярі	+	70%	10	5
3	Люди середнього віку з певним досвідом роботи в певній сфері, які перекваліфіковуються	+	100%	2	1
4	Ментори	+	98%	8	3

5	Працюючі люди з досвідом, які хочуть розвиватися допомагаючи іншим	+	100%	5	1
Які цільові групи обрано: всі описані групи					

Отже, обираю всі описані групи і буду використовувати стратегію масового маркетингу, адже всі групи зацікавлені в створюваному продукті і кожен з них знайде свою частинку функціоналу.

Для роботи в обраних сегментах ринку необхідно сформуємо базову стратегію розвитку (табл. 5.15).

Таблиця 5.2 – Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
	Працюємо із всім ринком, пропонуємо стандартизовану програму	Стратегія масовий маркетингу	Високий рівень автоматизації, задоволення потреб клієнтів і реалізацію різного функціоналу, абсолютно новий функціонал, якого ще немає на ринку, зручність інтерфейсу	Стратегія диференціації

Наступним кроком оберемо стратегії конкурентної поведінки (табл. 5.16).

Таблиця 5.3 – Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
-------	--	--	---	----------------------------------

1	Так	Шукатиме нових і забиратиме старих у існуючих конкурентів	Єдина характеристика, яка може бути скопійована – пошук ментора по імені. Вона буде скопійована	Стратегія лідера
---	-----	---	---	------------------

На основі вимог споживачів з обраних сегментів до постачальника (стартап-компанії) та до продукту (див. табл. 5.5), а також в залежності від обраної базової стратегії розвитку (табл. 5.15) та стратегії конкурентної поведінки (табл. 5.16) розробимо стратегію позиціонування (табл. 5.17). що полягає у формуванні ринкової позиції (комплексу асоціацій), за яким споживачі мають ідентифікувати торгівельну марку/проект.

Таблиця 5.4 – Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
1	Якість, високий рівень автоматизації, зручний інтерфейс, все в одному місці, новий функціонал: пошук	Стратегія диференціації	Високий рівень автоматизації, задоволення потреб клієнтів і реалізацію різного функціоналу,	Високий рівень автоматизації, задоволення потреб клієнтів, зручність інтерфейсу

	менторів-однодумців, введення кабінетів, розкладу та заявок, чат для спілкування		абсолютно новий функціонал, якого ще немає на ринку, зручність інтерфейсу	
--	--	--	---	--

Отже, для проекту більше підійде стратегія масового маркетингу, адже проект націлений на багато соціальних груп. Обираю стратегію диференціації як базову стратегію розвитку, адже саме ця стратегія якнайкраще підходить для програмного продукту, в якому багато нового функціоналу, який відповідатиме вимогам споживача. За стратегію конкурентної поведінки візьму стратегію лідерства, бо як такого аналогу на ринку не існує, і цей проект буде мати право називатися першим і бути лідером.

5.5 Розроблення маркетингової програми стартап-проекту

У розробленні маркетингової програми стартап-проекту першим кроком є формування маркетингової концепції продукту, який отримає споживач. Для цього у табл. 5.18 підсумуємо результати попереднього аналізу конкурентоспроможності товару.

Таблиця 5.5 – Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Якість, високий рівень автоматизації, зручний	Наявність нового функціоналу,	Високий рівень автоматизації, задоволення потреб клієнтів і реалізацію різного функціоналу, абсолютно новий

інтерфейс, все в одному місці, новий функціонал: пошук менторів-однодумців, введення кабінетів, розкладу та заявок, чат для спілкування	якого ще ніде немає	функціонал, якого ще немає на ринку, зручність інтерфейсу
---	---------------------	---

Розробимо трьохрівневу маркетингову модель товару: уточнюється ідея продукту та/або послуги, його фізичні складові, особливості процесу його надання (табл. 5.19).

Таблиця 5.6 – Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Рекомендаційна система для онлайн-освіти «EasyHelp»		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	1. Пошук менторів, що можуть бути потенційними однодумцями		Тл/Ор
	2. Кабінет ментора		Тл/Ор
	3. Заявки учнів		Тл/Ор
	4. Розклад онлайн		Тл/Ор
	5. Чат для спілкування		Тл/Ор

	Якість: Тестування за наявною документацією
	Пакування: відсутнє
	Марка: @Luda&Kate “EasyHelp”
III. Товар із підкріпленням	До продажу: гарантія якості
	Після продажу: підтримка
За рахунок чого потенційний товар буде захищено від копіювання: захист інтелектуальної власності	

Отже, бачимо, що захист продукту буде здійснюватися за рахунок права на інтелектуальну власність, а використані алгоритми будуть комерційною таємницею.

Наступним кроком визначимо цінові межі, якими необхідно керуватись при встановленні ціни на потенційний товар, яке передбачає аналіз ціни на товари-аналоги або товари субститути, а також аналіз рівня доходів цільової групи споживачів (табл. 5.20).

Таблиця 5.7 – Визначення меж встановлення ціни

№ п/п	Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1	Безкоштовно	Безкоштовно	Споживачі входять і групи з різноманітним рівнем доходів (бідні, середній клас, багаті, дуже багаті)	Безкоштовно

Насправді з наявних аналогів, лише деякі аналоги беруть відсоток з проведених занять. Більшість сайтів безкоштовні і вільні у доступі, тому

недоцільно робити сайт платним для користування. А щоб все-так проект був рентабельним і окупився, прибуток можна отримувати з реклами, яку можна буде розмістити у Web-застосуванні.

Формування системи збуту – немає збуту, оскільки Web-застосування у вільному користуванні.

Останньою складовою маркетингової програми є розроблення концепції маркетингових комунікацій, що спирається на попередньо обрану основу для позиціонування, визначену специфіку поведінки клієнтів (табл. 5.22).

Таблиця 5.8 – Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цілових клієнтів	Канали комунікацій, якими користуються цілові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1	Орієнтовані на якість та зручність	Контент- маркетинг, реклама	Новизна та якість функціоналу	Заохотити користуватися продуктом	Показати зручність і новизну продукції

Отже, користування Web-застосуванням безкоштовне, проект отримуватиме гроші за рекламу, яку буде там розмішувати. Web-застосування орієнтоване на різні соціальні верстви і не має обмежень у користуванні. Щоб заохотити споживачів користуватися продуктом буде задіяна реклама, завдання якої буде звернення уваги на даний продукт і його переваги, а саме буде пропагувати переваги нового онлайн-ресурсу.

Висновки. Існує потреба у створенні даного програмного продукту, оскільки наявний попит, а проект є рентабельним. Є перспективи впровадження даного проекту, оскільки проект є конкурентоспроможним та має багато сильних

сторін, що переважають над існуючими аналогами. Реалізуємо проект, шукаємо інвесторів, ведемо стратегію масового маркетингу, диференціації та лідерства на ринку як «першопрохідці».

Проект рентабельний і конкурентоспроможний, тож є цілком доцільним його подальше впровадження.

ВИСНОВКИ

Метою дисертаційної роботи є підвищення якості інформування потенційних споживачів та інтелектуалізація процесів надання освітніх послуг шляхом розробки оригінального програмно-алгоритмічного забезпечення та реалізації його у вигляді спеціалізованої програмної системи.

Мета досягнута шляхом виконання наступних завдань:

- виконати огляд існуючих постановок задач у сфері освіти;
- виконати огляд існуючих методів розв'язання задач складання розкладу;
- здійснити порівняльний аналіз різних методів та моделей та класифікувати їх;
- формалізувати задачу складання розкладу для менторів та учнів;
- розробити алгоритми локального пошуку і ОМК;
- розробити програмний продукт з реалізованими алгоритмами;
- виконати аналіз отриманих результатів.

Наукова новизна дисертації полягає у такому:

- запропоновано нову змістовну постановку задачі у сфері освіти;
- розроблено нова математична модель;
- розроблено та реалізовано програмно алгоритм локального пошуку для розв'язання даної задачі;
- розроблено та реалізовано програмно оптимізації мурашиними колоніями для розв'язання даної задачі;
- проведено дослідження ефективності запропонованих алгоритмів на основі аналізу результатів обчислювальних експериментів.

Отримані наукові результати покладені в основу розробленої спеціалізованої програмної системи інтелектуалізації процесів надання освітніх послуг онлайн, а також запропонованого відповідного стартап-проекту.

Подальші дослідження варто спрямувати на розробку модифікацій алгоритмів локального пошуку та мурашиних алгоритмів з метою підвищення точності отримуваних розв'язків. Важливим є дослідження впливу параметрів мурашиного алгоритму на його ефективність та час виконання. Можливі напрями модифікації:

- зміна стратегії вибору початкового варіанту розкладу;
- зміна критерію завершення;
- розширення та комбінація розроблених правил для алгоритму локального пошуку;
- зміна вхідних параметрів алгоритму оптимізації мурашиними колоніями.

Також одним із можливих напрямків досліджень є застосування інших методів комбінаторної оптимізації (GRASP, G-алгоритм), розробка алгоритмів і порівняння їх ефективності з уже розробленими алгоритмами.

ПЕРЕЛІК ПОСИЛАНЬ

1. Зінченко Л.В. До задач теорії розкладів у сфері освіти // Polish journal of science. – 2019. – №16. – С. 37.
2. Зак Ю.А. Прикладные задачи теории расписаний и маршрутизации перевозок / Ю.А. Зак. – М.: Книжный дом «ЛИБРОКОМ», 2012. – С. 394.
3. Сперкач М.О. Інформаційна технологія оперативно-календарного планування дрібносерійного виробництва за концепцією «точно в строк» : автореф. дис. ... канд. техн. наук. : 05.13.06 – інформаційні технології / Майя Олегівна Сперкач. – Київ, 2016. – С. 39 .
4. Бойко О.М. Еволюційна теорія розв'язування задачі складання розкладів навчальних занять // Штучний інтелект. – 2006. – № 3. – С. 341-348.
5. Бевз С. В. Розробка автоматизованої системи формування розкладу магістратури / С. В. Бевз, В. В. Войтко, С. М. Бурбело, А. М. Шоботенко // Наукові праці ВНТУ. – 2008. – № 4. – С. 13.
6. Безгинов А. Н. Комплекс алгоритмов построения расписания вуза. Ч. 1 : Система оценки качества расписания на основе нечетких множеств, алгоритм поиска оптимального расписания / А. Н. Безгинов, С. Ю. Трегубов // Вестник Балтийского федерального университета им. И. Канта. – 2011. – Выпуск 5. – С. 127-135.
7. Бабкина Т. С. Задача составления расписаний: решение на основе многоагентного подхода / Т. С. Бабкина // Бизнес-информатика. – 2008. - №1. – С. 23-28.
8. Шостак И. В. Автоматизация процесса составления расписания занятий на основе тензорного исчисления в учебном комплексе / И. В. Шостак, К. Э. Яновская, С. В. Россоха // Авиационно-космическая техника и технология. – 2012. – №9. – С. 263-266.
9. Ризун Н. О. Применение методов декомпозиции при решении многокритериальной задачи автоматизации составления расписания учебных

занятий в ВУЗе / Н. О. Ризун // Східно-Європейський журнал передових технологій. – 2012. – № 2. – Issue 4. – С. 59-70.

10. Кабальнов Ю. С. Композиционный генетический алгоритм составления расписания учебных занятий / Ю. С. Кабальнов, Л. И. Шехтман, Г. Ф. Низамова, Н. А. Земченкова // Вестник Уфимского государственного авиационного технического университета. – 2006. – Т. 7. – №2. – С. 99-107.

11. Конькова И. С. Генетические алгоритмы в решении задачи составления расписания занятий в вузе / И. С. Конькова // Проблемы информатики в образовании, управлении, экономике и технике : Сб. Стетей XII Междунар. научно-техн. конф. – Пенза : ПДЗ, 2012. – С. 26-29.

12. Лебединская Е. Н. Формализация задачи построения рационального расписания занятий в системе АСУ ВУЗ / Е. Н. Лебединская, Е. Г. Шепилова, С. А. Хатламаджиян // Наука и техника транспорта. – 2003. – №1. – С. 67-73.

13. Попов Г. А. Формализация задачи составления расписания в высшем учебном заведении / Г. А. Попов // Вестник АЕТУ – 2006. – №1. – С. 118-139.

14. Гуляницький Л.Ф., Мулеса О.Ю. Прикладні методи комбінаторної оптимізації: навч.посібник. – К.: Видавничо-поліграфічний центр «Київський університет», 2016. – С. 242.

15. Танаев В.С., Шкурба В.В. Введение в теорию расписаний. – М.: Наука, 1975. – С. 256.

16. Зінченко Л. В. About the problems of the schedule theory in education // Innovative solutions in modern science. – 2019. – №6(33). – С. 48-70.

17. Talbi E-G. Metaheuristics: from design to implementation // John Wiley & Sons. – 2009. – № 74. – С. 24.

18. Rozanski N. Woods E. Software Systems Architecture: Working with Stakeholders Using Viewpoints and Perspectives. – 2008. – С. 345.

19. Руководство по ASP.NET Core 3 [Электронный ресурс]/Режим доступа: <https://metanit.com/sharp/aspnet5/>.

20. Брила А.Ю., Антосяк П.П., Глебена М.І., Чупов С.В., Семейон І.В. Основи програмування у С#. – Ужгород, 2014. – С. 60.
21. Microsoft Visual Studio [Електронний ресурс]/Режим доступу: http://db0nus869y26v.cloudfront.net/uk/Microsoft_Visual_Studio.
22. Преимущества высочайшей производительности и надежной защиты SQL Server 2017 [Електронний ресурс]/Режим доступу: <https://itpro.ua/product/microsoft-sql-server-2017/?tab=description>.
23. Gamma E., Helm R., Johnson R., Vlissides J. Design Patterns: Elements of Reusable Object-Oriented Software. – Addison-Wesley, 1994. – pp. 237.
24. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. — Приемы объектно-ориентированного проектирования. Паттерны проектирования. – 2001. – С. 89.
25. LINQ to SQL [Електронний ресурс]/Режим доступу: <https://docs.microsoft.com/ru-ru/dotnet/framework/data/adonet/sql/linq/>.
26. Файн Я., Моисеев А. Angular и TypeScript. Сайтостроение для профессионалов. – СПб.: Питер, 2018. — С. 464.
27. Node.js [Електронний ресурс]/Режим доступу: <https://nodejs.org/>.
28. Обзор 4-й версии Bootstrap: описание, преимущества и недостатки [Електронний ресурс]/Режим доступу: <https://tokar.ua/read/8707>.
29. Font Awesome 5 Free [Електронний ресурс]/Режим доступу: <https://github.com/FortAwesome/Font-Awesome>.

ДОДАТОК А

Графічний матеріал

СТРУКТУРА ТЕРМІНІВ ДЛЯ СФЕРИ НАВЧАННЯ



Демонстраційний плакат до магістерської дисертації
на тему «Інформаційна рекомендаційна система в сфері освітніх послуг»

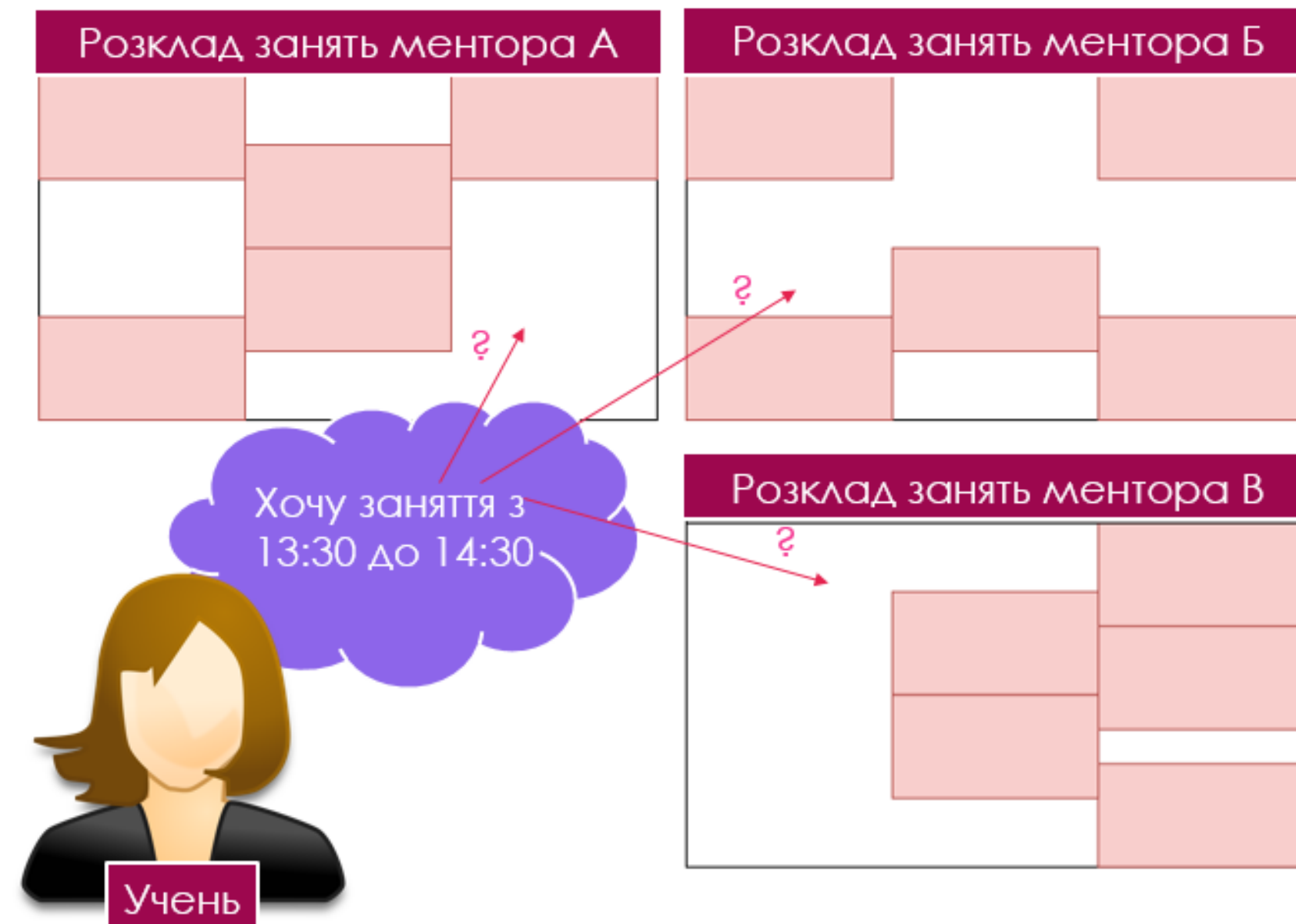
Виконала студентка гр. ІС-82мп

Зінченко Людмила Вікторівна

Керівник

Гуляницький Леонід Федорович

ВІЗУАЛІЗАЦІЯ ПОСТАНОВКИ ЗАДАЧІ



Демонстраційний плакат до магістерської дисертації
на тему «Інформаційна рекомендаційна система в сфері освітніх послуг»

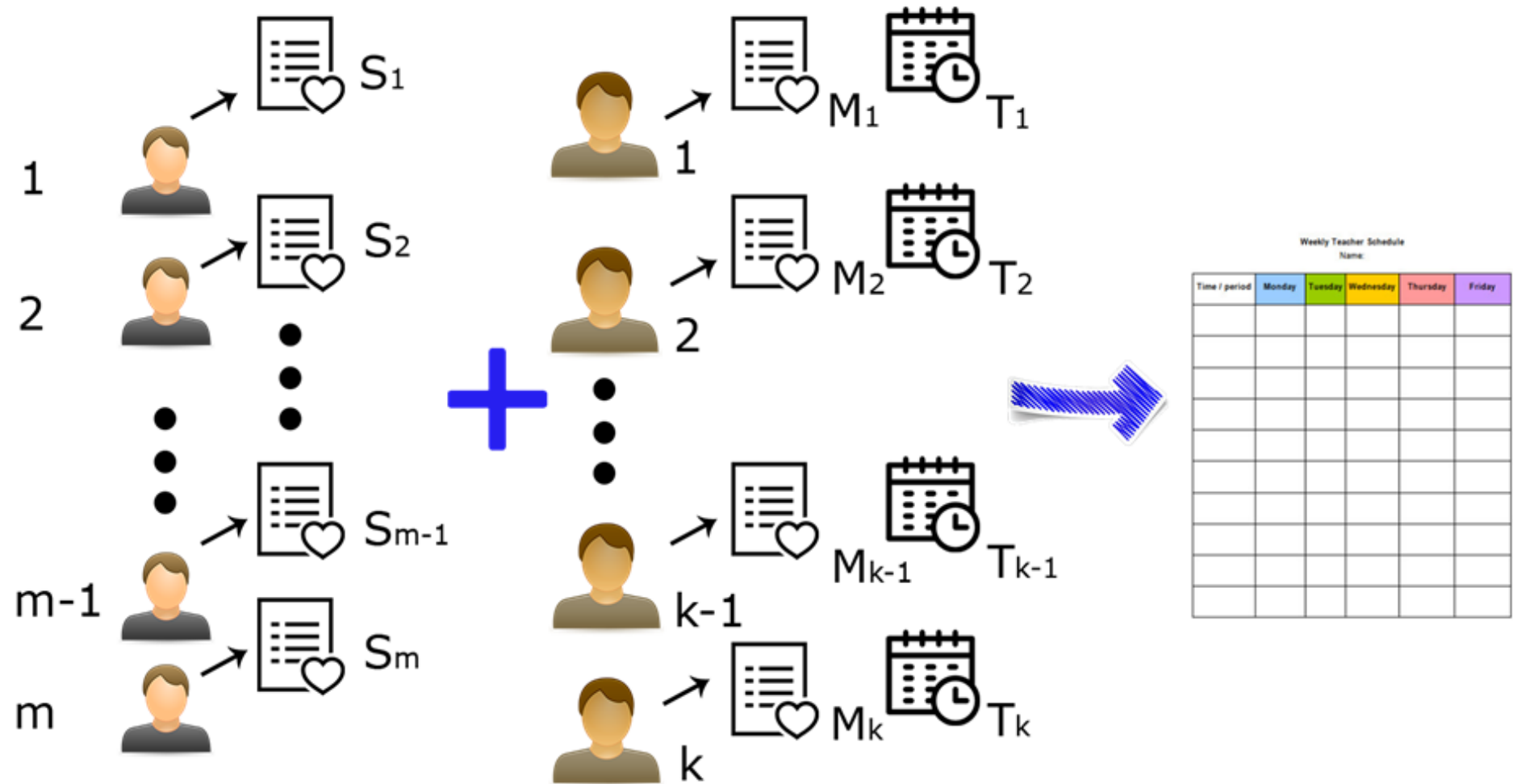
Виконала студентка гр. ІС-82мп

Зінченко Людмила Вікторівна

Керівник

Гуляницький Леонід Федорович

СХЕМАТИЧНЕ ЗОБРАЖЕННЯ ВХІДНИХ І ВИХІДНИХ ДАНИХ ЗАДАЧІ



Демонстраційний плакат до магістерської дисертації

на тему «Інформаційна рекомендаційна система в сфері освітніх послуг»

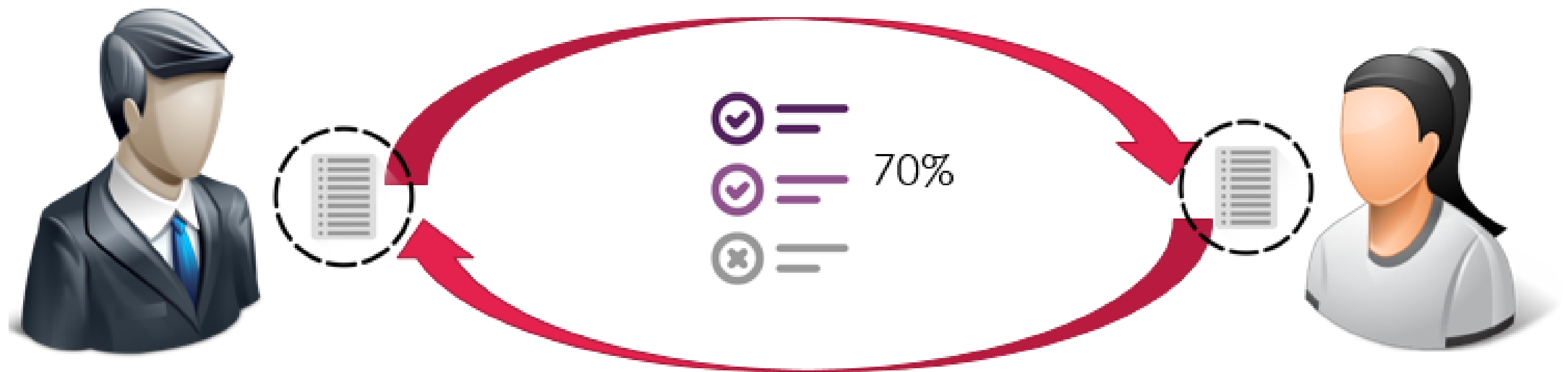
Виконала студентка гр. ІС-82мп

Зінченко Людмила Вікторівна

Керівник

Гуляницький Леонід Федорович

ПРОЦЕС ВИЗНАЧЕННЯ ОДНОДУМЦІВ



Демонстраційний плакат до магістерської дисертації
на тему «Інформаційна рекомендаційна система в сфері освітніх послуг»

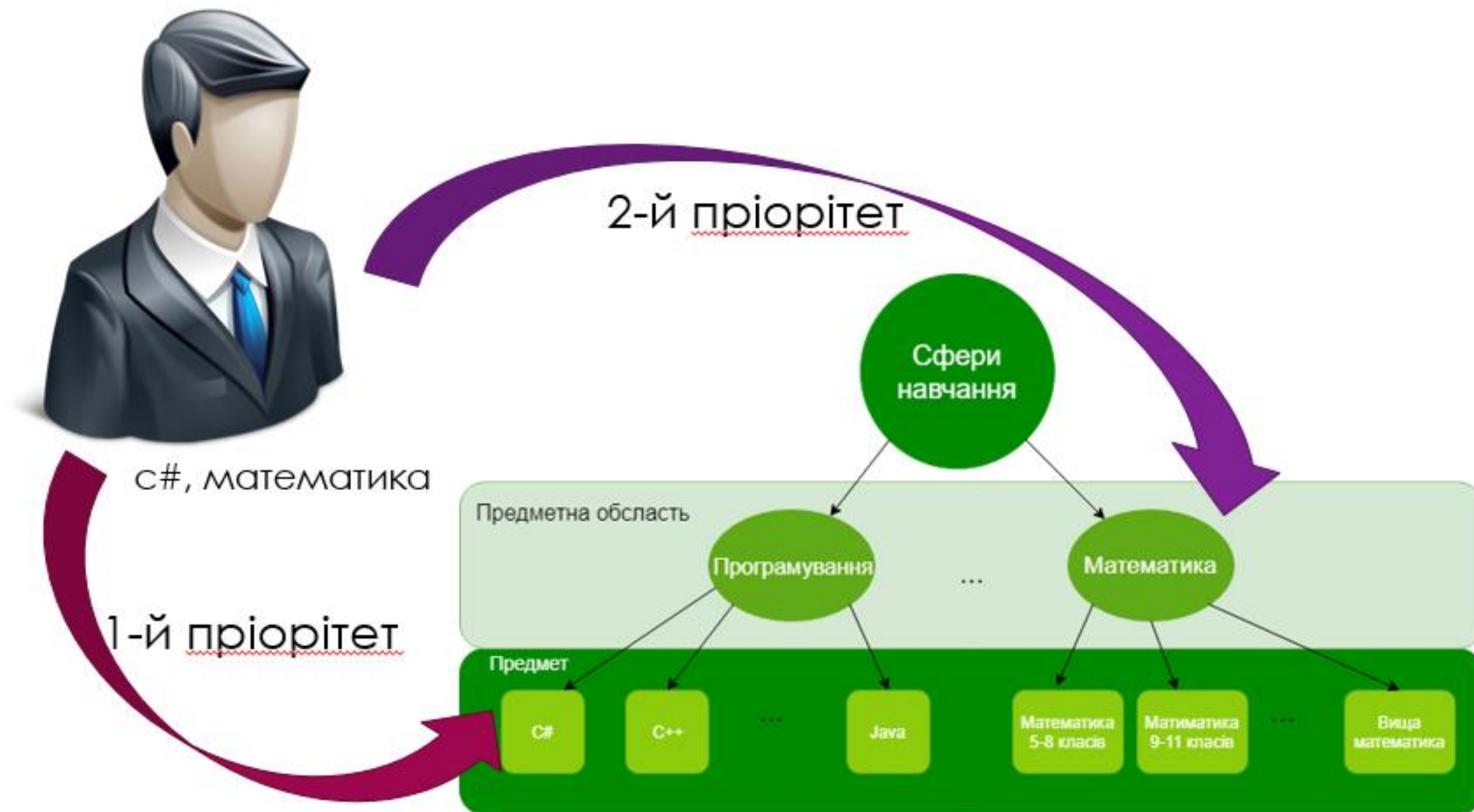
Виконала студентка гр. ІС-82мп

Зінченко Людмила Вікторівна

Керівник

Гуляницький Леонід Федорович

ПРОЦЕС ВИЗНАЧЕННЯ ВАЖЛИВОСТІ ПРЕДМЕТНОЇ ОБЛАСТІ



Демонстраційний плакат до магістерської дисертації
на тему «Інформаційна рекомендаційна система в сфері освітніх послуг»

Виконала студентка гр. ІС-82мп

Зінченко Людмила Вікторівна

Керівник

Гуляницький Леонід Федорович

ПСЕВДОКОД АЛГОРИТМУ ЗНАХОДЖЕННЯ ОДНОГО ВАРІАНТУ РОЗКЛАДУ

$F := 0$

for $j = 1, \dots, M$ do { j – ментор }

for $t = 1, \dots, T$ do { x_{tj} – номер заявки }

Беремо схожість ментора j з учнем $c_{1,x_{tj}}$:

$$v_{c_{1,x_{tj}}j}$$

Беремо важливість для ментора j предметної області $c_{2,x_{tj}}$:

$$w_{c_{2,x_{tj}}j}$$

$$F := F + \alpha(1 - v_{c_{1,x_{tj}}j}) + (1 - \alpha)(1 - w_{c_{2,x_{tj}}j})$$

endfor t

endfor j

Демонстраційний плакат до магістерської дисертації

на тему «Інформаційна рекомендаційна система в сфері освітніх послуг»

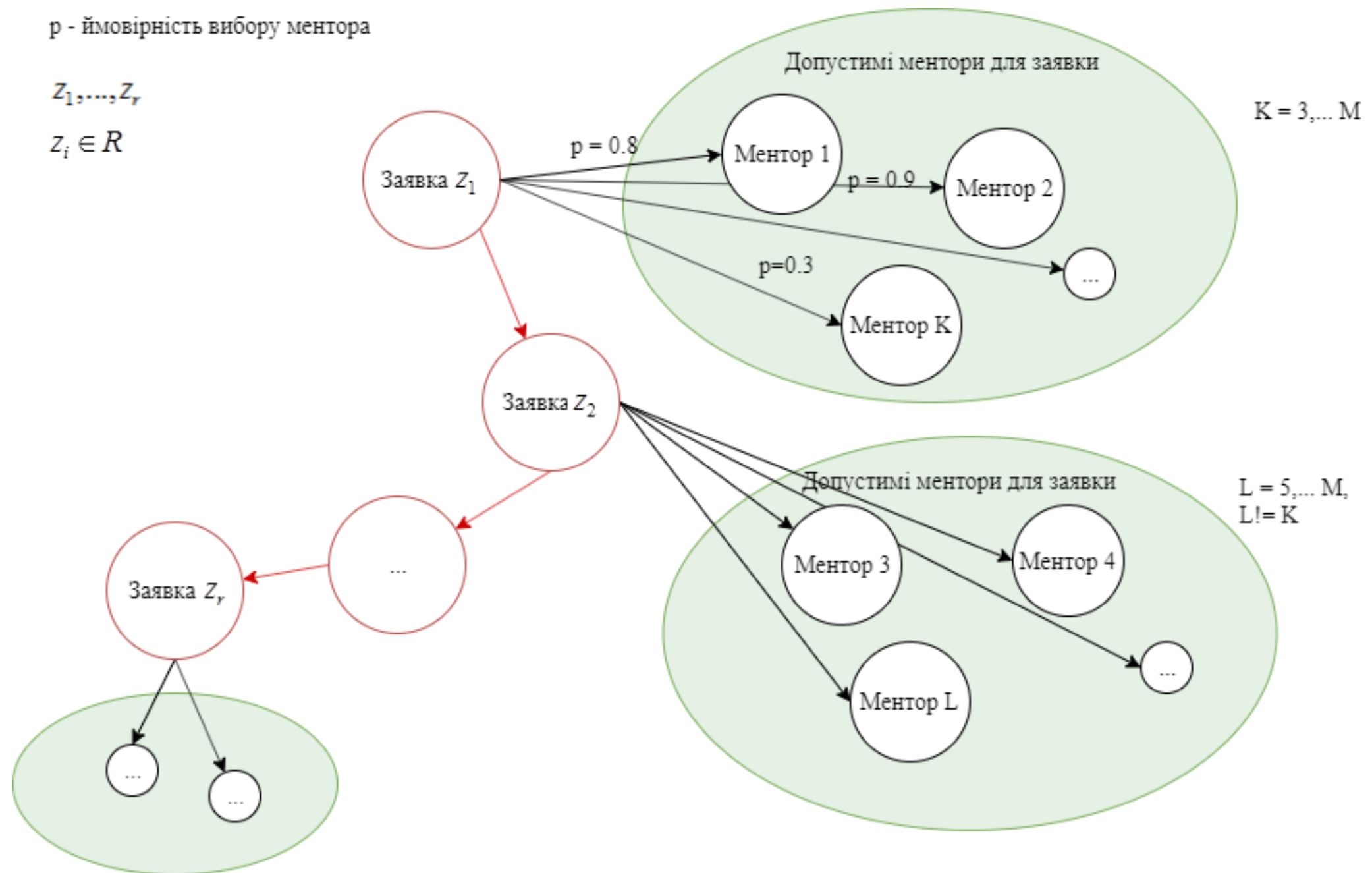
Виконала студентка гр. ІС-82мп

Зінченко Людмила Вікторівна

Керівник

Гуляницький Леонід Федорович

СТРУКТУРА ГРАФУ ДАНОЇ ЗАДАЧІ ДЛЯ МУРАШИНОГО АЛГОРИТМУ



Демонстраційний плакат до магістерської дисертації
на тему «Інформаційна рекомендаційна система в сфері освітніх послуг»

Виконала студентка гр. ІС-82мп

Зінченко Людмила Вікторівна

Керівник

Гуляницький Леонід Федорович